

Article

Visual Flight Rules-Based Collision Avoidance Systems for UAV Flying in Civil Aerospace

Hamid Alturbeh ^{1,*} and James F. Whidborne ²¹ Institute of Railway Research, University of Huddersfield, Huddersfield HD1 3DH, UK² Centre for Aeronautics, Cranfield University, Cranfield MK43 0AL, UK; j.f.whidborne@cranfield.ac.uk

* Correspondence: h.alturbeh@hud.ac.uk

Received: 5 December 2019; Accepted: 20 February 2020; Published: 25 February 2020



Abstract: The operation of Unmanned Aerial Vehicles (UAVs) in civil airspace is restricted by the aviation authorities, which require full compliance with regulations that apply for manned aircraft. This paper proposes control algorithms for a collision avoidance system that can be used as an advisory system or a guidance system for UAVs that are flying in civil airspace under visual flight rules. A decision-making system for collision avoidance is developed based on the rules of the air. The proposed architecture of the decision-making system is engineered to be implementable in both manned aircraft and UAVs to perform different tasks ranging from collision detection to a safe avoidance manoeuvre initiation. Avoidance manoeuvres that are compliant with the rules of the air are proposed based on pilot suggestions for a subset of possible collision scenarios. The proposed avoidance manoeuvres are parameterized using a geometric approach. An optimal collision avoidance algorithm is developed for real-time local trajectory planning. Essentially, a finite-horizon optimal control problem is periodically solved in real-time hence updating the aircraft trajectory to avoid obstacles and track a predefined trajectory. The optimal control problem is formulated in output space, and parameterized by using B-splines. Then the optimal designed outputs are mapped into control inputs of the system by using the inverse dynamics of a fixed wing aircraft.

Keywords: unmanned aerial vehicles; collision avoidance system; optimal control system; trajectory planning; B-splines

1. Introduction

Unmanned Aircraft Systems (UAS) are being increasingly used for both civilian and military applications particularly due to their ability to complete dull, dirty and dangerous missions [1]. The operation of Unmanned Aerial Vehicles (UAVs) in civil/non-segregated airspace is, however, restricted by the policies and regulations of the aviation authorities (e.g., Civil Aviation Authority (CAA) in the UK, Federal Aviation Administration (FAA) in the USA), which require full compliance with the rules and obligations that apply for manned aircraft [2–4].

In a manned aircraft the pilot in command has the ultimate responsibility for achieving the collision avoidance manoeuvre using the see and avoid principle. A pilot's decision-making process during a conflict can be broken down using the so-called Observe, Orient, Decide, and Act loop [3,5]. The required time for a pilot to recognize an approaching aircraft and initiate an avoidance manoeuvre is around 12.5 s [6] but may be greater because pilots differ in their response time [7]. Most of this time is spent on collision recognition and decision making. Hence, a Decision-Making System (DMS) that could be used as an advisory system will effectively save time and help both the on-board pilot in manned aircraft, and the ground-based pilot for UAV's to avoid the conflicts safely. In an autonomously operating UAV, a DMS could be used to initiate avoidance manoeuvres.

This paper presents control algorithms for a collision avoidance system for UAVs operating in civil airspace under visual flight rules. The main contributions can be summarized as follows:

- Development of a DMS architecture for collision avoidance system in VFR conditions. The proposed DMS architecture mimics the pilot decision making process during a conflict scenario.
- Propose, construct, and parameterize collision avoidance manoeuvres for a set of conflict scenarios:
 1. Head-on/overtaking conflict scenarios.
 2. Approaching conflict scenarios.

The collision avoidance manoeuvres are proposed based on pilot suggestions (Extended interviews and discussions about the problem have been carried out with a pilot at the National Flying Laboratory Centre, Cranfield University). Hence, the shapes of the manoeuvres are similar to those performed by manned aircraft.

- A geometric approach is proposed to parameterize the generated collision avoidance manoeuvres. Thus the construction and generation the avoidance manoeuvres are simplified, hence the computational time for avoidance manoeuvre generation is reduced.
- Development of a real-time local trajectory planning algorithm for a fixed-wing UAV using B-splines and Model Predictive Control (MPC) system. The developed method is an extension of a previous method that was proposed for a quad-rotor UAV [8]. Inverse dynamic model of the fixed-wing aircraft has been developed using the differential flatness property of the fixed-wing aircraft. The inverse dynamic model is used to map the generated trajectory into the UAV's control commands. A method that helps the optimization solver to avoid a local minimum is proposed. Note that some of this section has been presented previously [9].

Note that the paper is not intended as a full prototype system. The obstacle detection problem, which is a significant part of a full system and is a major research challenge in itself, is not considered in this paper. Instead it is assumed that the needed parameters for the UAV and obstacles/intruders are available for the proposed algorithm. The paper practical contribution is in terms of the integration with the Rules-of-the-air. Note also that the system is not limited to UAVs, but is applicable to UAVs, remotely piloted UAVs and manned aircraft. As such, the hardware implementation is not within the scope of the paper.

This paper consists of nine sections including the introduction section. The remainder of this paper is structured as follows. Section 2 provides a literature review of trajectory planning method for collision avoidance systems for aircraft. Section 3 discusses the requirements and regulations of using the UAVs in civil airspace. Section 4 proposes a DMS algorithm for the Collision Avoidance Systems (CAS). Section 5 discusses the avoidance manoeuvres trajectory profiles generating process for different conflict scenarios in which the UAV should change direction (right/left turn) in horizontal plane. Section 6 shows some simulation results of the proposed system for a set of conflict scenarios. Section 7 presents a local trajectory planning algorithm that is used for the collision avoidance system for a fixed-wing aircraft. Section 8 tests the developed local trajectory planning algorithm and presents the simulation results. The conclusions are given in Section 9.

2. Collision Avoidance Systems (CAS) and Trajectory Planning

In recent years, developments in sensors technology and powerful processing units have led to a significant enhancement in both detection and resolving conflict scenarios [10]. References [11,12] have categorized the collision avoidance approaches based on fundamental factors that can express and identify the differences between each method, such as; sensing tools, encounter sensing dimension, encounter current states projection, collision threat assessment, avoidance trajectories calculation, and manoeuvre realization. Many approaches have been proposed to find an adequate solution for the collision avoidance problem such as; Predefined Collision Avoidance [13], Protocol Based Decentralized Collision Avoidance [14–17], Optimized Escape Trajectory Approaches [18–22], Potential

Field Methods [23], Geometric Methods [24–27], and other approaches include trajectory estimators and hybrid CAS systems [14,28].

A trajectory planner can be categorized as one of two types [29]: a global planner which requires a good knowledge about the environment that the aircraft is going to fly in, and a local trajectory planner which is an algorithm that is running continuously in order to allow the aircraft to deal with events that may happen during the flight. Figure 1 shows a simple flight scenario where the aircraft mission is to fly from point A to B with the existence of both a pre-known obstacle and an intruder which is unknown till the sensing devices detect it during the journey.

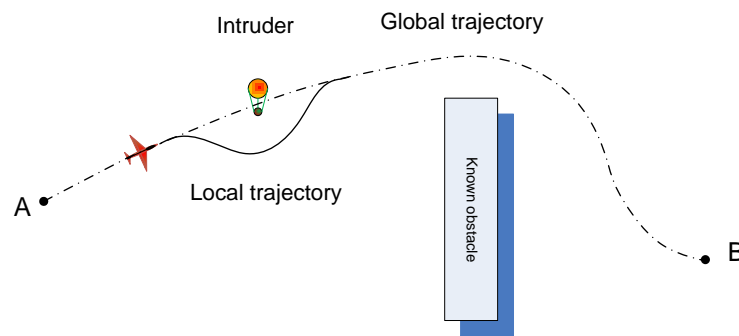


Figure 1. Simple flight scenario: global and local trajectories.

In order to complete this mission successfully, the global planner will calculate the optimal trajectory for whole journey from A to B taking into consideration all known obstacles. The local trajectory planner tracks the global trajectory and avoids the detected obstacle. After resolving the conflict the aircraft will continue tracking the global trajectory heading to the destination point B.

There are many algorithms and approaches that have been proposed to solve the problem of local trajectory planning, including the potential field method, roadmap method, approximate and exact cell decomposition, rapidly exploring random trees method and wavefront planning method [30,31], some of which are summarized below. The number of dimensions, the description of the constraints and the space configuration contributes to the numerous factors that decide the most suitable method for a given problem of planning. As the difficulty increases with the number of dimensions, most of the approaches are restricted to two dimensional (2D) spaces.

2.1. Potential Field Method

This method was first presented by Khatib [23] for robotics. It expresses the way-points as attractive forces and the obstacles as repulsive forces. By using simple electrostatic equations, a safe trajectory can be generated and then the trajectory with a low flux density is selected as the preferred path. This approach is appropriate for distributed and local collision avoidance where state information is available from all aircraft and when the number of vehicles are small [32]. However, many difficulties arise in practical systems such as saddle points and local minima that may occur when generating a dynamic potential field and this may lead to aircraft loss of control or collision threat.

Another problem that may be faced in a practical implementation of this method is that the dynamic limitations of the aircraft are not considered. Hence, the aircraft may not be able to fly the generated trajectory. Finally, it is worth mentioning that the availability of state information is an essential factor for potential field method. So any deficiency in the state information may produce an improper field formation and then generate an aggressive control command that may be beyond the aircraft performance [33].

2.2. Wavefront Planning Method

The wavefront planning method breaks the region into pieces or cells in a framework and is also identified as numerical potential field method [34]. Then the cells are given a numbers such that the

free cells are given zeros, the cells with obstacle are given ones, and the cell with target is given the value of two. Then a wavefront from the target will be formed by setting the neighbouring cells around target that have zero-valued with three and then setting the zero-valued cells adjacent to threes with four and so on. The process stops when the unit comprising start is reached by the wavefront. the path can be found by using a gradient descent from the start cell. In order to avoid the local minimum randomized planning approaches [35–37] have been proposed by starting a sequence of arbitrary walks when immovable.

2.3. Sample-Based Approaches

This method starts by creating a map that can later be explored for a collision free path. A map is created using Probabilistic Road-maps [38] by sampling points in the network of space. Nodes will be then added to the map at the points that are outside the obstacles. Then the nodes are used to link the network that near each other in order to find free collision paths. The best free collision path can be identified on a map by using traditional approaches include Dijkstra's algorithm [39] or A* [40]. However, the vehicle dynamic limitations make it difficult to locate the networks between the samples. Rapidly Exploring Random Trees is continuation of Probabilistic Road-maps method which includes vehicle dynamics, hence it takes into consideration the dynamic limitation. With Rapidly Exploring Random Trees, a tree is repetitively extended by using control inputs that induce the system vaguely near randomly chosen points [41–43].

3. UAV Integration in Civil Airspace

In order to integrate the UAV in the civil airspace it must meet an Equivalent Level Of Safety (ELOS) comparable to a manned aircraft. ELOS refers to a combination of systems and a concept of operations that reduce the chance of midair collision to an acceptable level [44]. Two groups are leading the development of standards for safe and transparent UAS integration into non-segregated airspace: EUROCAE WG-73 in Europe and RTCA SC-203 in the US [45]. A comparative study for these groups' activities can be found in [45]. Some civil aviation authorities have published general requirements for UAV operations in civil airspace, such as CAP-722 (Unmanned Aircraft System Operations in UK Airspace-Guidance) that published by the CAA [46], and FAA road-map for the integration of civil UAS in the National Airspace System [2]. Reference [44] establishes the requirements for a sense-and-avoid system for a remotely piloted aircraft that fulfills the intent of collision avoidance contained in the United States Federal Aviation Regulations and the convention on international aviation rules of the air.

Guidelines for the safety integration of autonomous UAS into civil airspace has been proposed in [47], which presents the challenges and proposes solutions to overcome them by both the UAV manufacturers/operators and the civil aviation organisations. Reference [48] reviews the manned aviation policies, procedures, and requirements and relates them to the equivalent policies and requirements for unmanned aviation. Many projects have been conducted to achieve the civil aviation authorities' requirements for UAS integration in all classes of the civil airspace. The following are some examples of these projects:

- Mid Air Collision Avoidance System (MIDCAS) (2009–2014) aimed to demonstrate the baseline of acceptable solutions for the critical UAS self separation and midair collision avoidance functions to contribute to the UAS integration in civilian airspace [4].
- Autonomous System Technology Related Airborne Evaluation and Assessment (ASTRAEA) focused on the technologies, systems, facilities, procedures and regulations that will allow autonomous vehicles to operate safely and routinely in civil airspace over the United Kingdom [3].
- Sense and Avoid Flight Tests (SAAFT), sponsored by the Air Force Research Laboratory (AFRL), which established the SAAFT program to demonstrate autonomous collision avoidance capabilities in both cooperative and non-cooperative air traffic. The intent of the program is to

equip UAVs with collision avoidance capabilities and thus allow them the same access to national and international airspace that manned aircraft have [49].

Although most research programmes focus on the technical requirements for UAV integration in the civil airspace, recently legal and ethical questions for using UAVs in non-segregated airspace have raised. Dubot [50] has proposed the first set of laws that should be applicable to unmanned aircraft operating autonomously.

This research aims to develop a collision avoidance system that is able to issue the resolution advisories, and generate and track safe avoidance manoeuvres. These manoeuvres should be similar to those performed by a pilot in manned aircraft which are compliant with the rules of the air.

4. Decision-Making System Based on The Rules of the Air

A DMS for collision avoidance is developed based on the rules of the air in Visual Flight Rules (VFR) conditions and CAA requirements [46]. The proposed architecture of the DMS is engineered to be implementable in manned aircraft to perform different tasks ranging from collision detection to generating avoidance Resolution Advisories (RA), and unmanned aircraft (remotely piloted and autonomous UAV). The DMS is divided into multiple layers, where each layer is built to perform a specific function. Figure 2 shows the proposed DMS architecture:

- Layer-1 Collision Detection and Prioritizing (CDP): This comprises two subsystems:
 - Collision Detection (CD): Detects the conflicts and generates alerts and warnings flags that will be used by other layers.
 - Risks Prioritizing (RP): Defines which of the detected intruders has a greater threats than the others, hence gives it a higher priority.
- Layer-2 Collision Assessment (CA): Detriment the collision scenario type (i.e., head-on, left/right approaching, and overtaking/overtaken).
- Layer-3 Advisory System (ADS): Gives general conflict resolution advisories (e.g., turn right, turn left, climb, descend, level, and hold speed and altitude).
- Layer-4 Avoidance Manoeuvre Generator (AMG): Generates avoidance manoeuvre that allows the aircraft to avoid the collision.

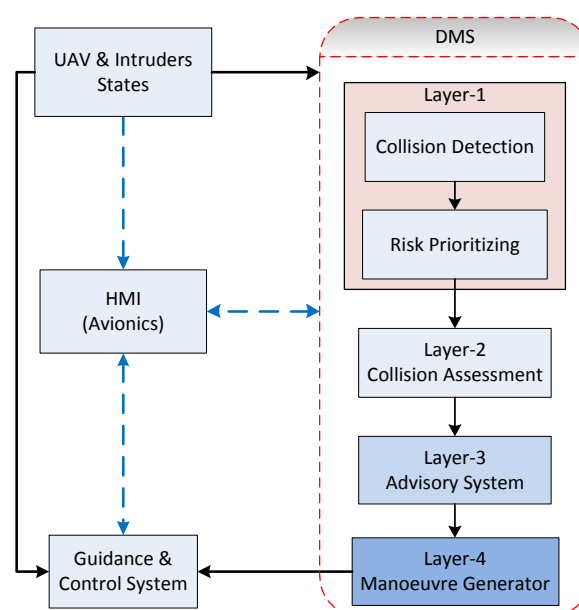


Figure 2. Decision-Making System (DMS) architecture.

The proposed DMS architecture is engineered to be implemented for different functionalities for manned aircraft, and at different level of autonomy of UAS, or at different classes of the flight control mode (Class-0 to Class-6) [46]. For example, in Class-0 or manned aircraft the DMS can be used to reduce the overall pilot workload by performing the following tasks:

- Collisions detection and risk prioritizing (Layer-1), so the pilot will know where the traffic is.
- Determines the collision scenario type (Layer-1, and Layer-2) this will help the pilot to decide what actions are needed to avoid the conflict.
- Generates conflict resolution advisories by evaluating the collision type (Layer-3) to help the pilot to initiate a suitable avoidance manoeuvre.

For the remotely piloted UAV (Class-1) this architecture helps the remote pilot to be aware of the surrounding conflicts so the remote pilot could act in the same manner as an on-board pilot. Moreover, Layer-4 can be used to activate highly autonomous mode to generate the avoidance manoeuvre. Layer-4 can be used also in flight mode classes (2–6). The next subsections give a description of the DMS layers.

4.1. Collision Detection and Risk Prioritizing Layer

This layer generates Alert Flags (AF1, AF2), Collision Flag (CF), and gives general information about the collision risks, such as the intruder direction.

The inputs of the collision detection layer are the range, the range rate, the relative altitude, and the bearing rate. The outputs are:

- Separation alert flags (AF1, AF2): Activated for the aircraft (intruder) which their range and relative altitude are less than specific values (loss of separation). Two different sets of ranges and relative altitudes can be used to generate two types of alerts. For example, AF1, is activated when intruder within (range R_1 , and relative altitude L_1) and AF2 will be activated for aircraft within (range R_2 , and relative altitude L_2). Having two levels of alert may help increase the safety and may help the pilot to prioritize the intruders in case of multi-intruder scenarios. Some commercial Portable Collision Avoidance Systems (PCAS) like XRX [51] give the user the ability to control the values of the range and the relative altitude. For example, the range can be selected to be 6 NM, 3 NM, or 1.5 NM (1 NM = 1852 m), and the relative altitude can be ± 2500 feet, ± 1500 feet, or ± 500 feet (1 foot = 0.3048 m).
- Collision flag (CF): Activated when a collision risk is detected by monitoring the the bearing angle (if bearing angle is constant then the collision risk is activated).

Prioritizing

This is very important for multi-intruder scenarios as it defines which intruder should be given a higher priority than the others so it can be avoided first. The proposed algorithm uses two different methods for intruder prioritizing:

- The first method is a modified version the commercial (PCAS) XRX [51]. However, the commercial PCAS is designed for manned aircraft, where the Communication and Control (C2), and data link problems and delay problem are not issue as same as the remotely controlled UAV [2]. Remotely piloted UAV needs a greater safety margin, hence three levels of alerts and warning (AF1, AF2, CF) are proposed in this paper to give the remote pilot more time and greater safety margin which would overcome the problems that may be associated with C2 link. The remote pilot can modify the values of ranges and relative altitudes depend on the UAV manoeuvrability and the flight environment. The prioritizing process can simply be determined by: intruder range and relative altitude, intruder vertical speed (climbing or descending), and aircraft vertical speed (climbing or descending). The prioritizing algorithm gives the highest priority to the intruder with (CF) flag and gives the lowest priority to the intruder with (AF1) flag. Intruders that share

the same flag will be prioritized based on their relative altitude from the aircraft, so the lower the vertical separation the higher the priority. However, if the aircraft is descending/climbing then the relative altitude sign will be taken in consideration. For example, if there are two intruders in level flight and both are within $\pm L_2$ and both have (AF2) flag, the intruder at higher level will be given higher priority if the aircraft is climbing, but it will given lower priority if the aircraft is descending.

- Using Time to Collision (T_c): One drawback of the previous method for prioritizing is that the relative speeds of the intruders with the aircraft are not taken into consideration. The relative speed between the aircraft and an intruder is function to their speeds and headings. Thus, using the ranges alone is not enough for the prioritizing process. In this paper time to collision (T_c) is used in prioritizing process, the smaller the T_c , the higher the priority will be given. The range is implicit in the time to collision ($T_c = R_a / V_r$, where R_a is the range, and V_r is the relative speed or closure rate $V_r = d(R_a) / dt$).

4.2. Collision Assessment Layer

This layer receives the collision information of the highly prioritized intruder and determines the conflict type. The assessment process uses heading angles of the aircraft and intruder, and the bearing angle as inputs then applying the right of the way and the rules of the air in see and avoid environment. The rules of right of way around the aircraft are shown in Figure 3. It can be seen that the aircraft must give way to all traffic approaching from the opposite direction within 45° of the aircraft's centreline and for all air traffic that are approaching from starboard (right) side of the aircraft. Balloons, gliders, airships, and aircraft towing objects always have the right of way except for cases when aircraft are being overtaken by them. An aircraft has right of way when it is overtaken by an aircraft within 70° of the aircraft's centreline [52].

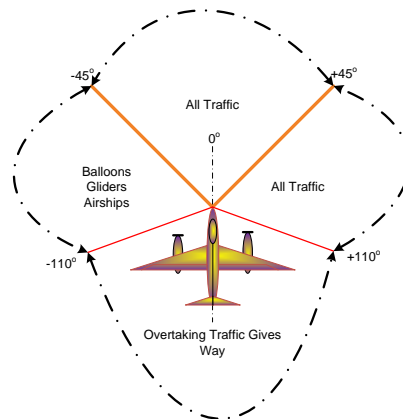


Figure 3. Rules of right of way around the aircraft.

Figure 4 shows a simple illustration of the three main collision scenarios with their collision avoidance rules. Both aircraft should turn right to avoid collision in head-on case. In the converging scenario, aircraft (A) must turn right to give way to aircraft (B) then tracks behind it. In the overtaking case, the overtaking aircraft (A) must turn right to keep out of the way of the overtaken one (B) that will keep going on its way [53,54]. These cases are the general cases and there are many sub-scenarios under each case.

Figure 5 shows the flowchart of the assessment algorithm, where the bearing and relative speed are used to assess the type of conflict.

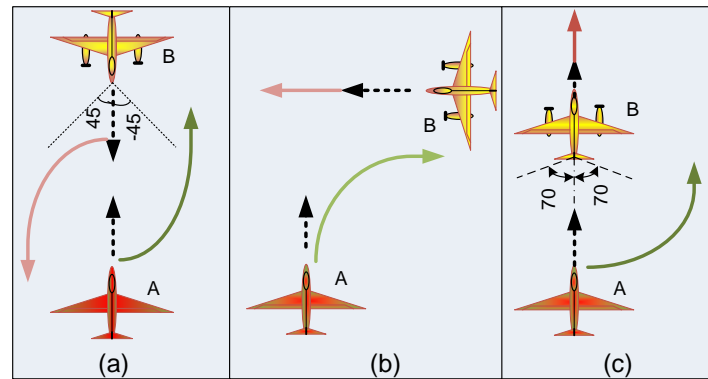


Figure 4. Main collision scenarios and avoidance rules: (a) head-on, (b) converging, and (c) overtaking.

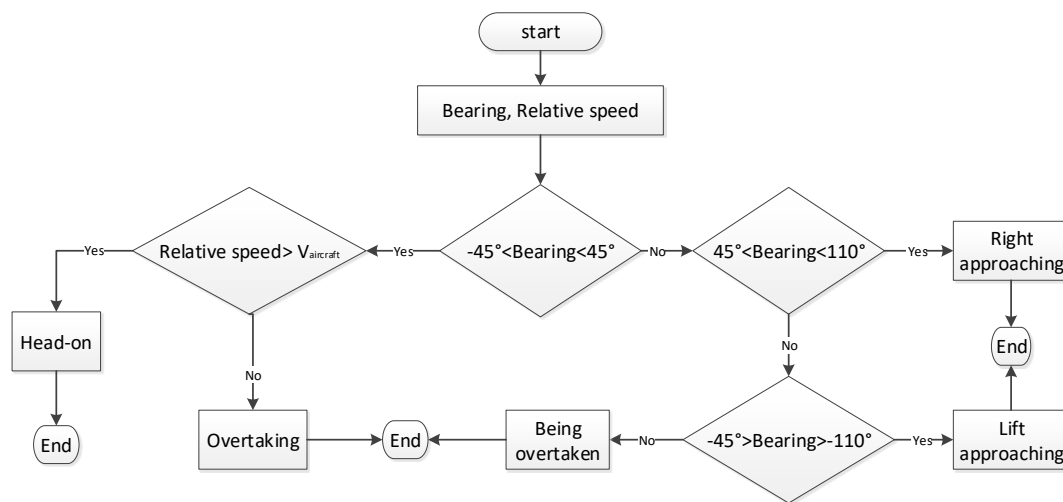


Figure 5. Collision assessment algorithm flowchart.

The assessment output (collision type, the relative speed or closure rate V_r , and the time to collide T_c) can be then passed to the pilot (in manned aircraft), or the remote pilot in case of UAV. The collision assessment layer also generates two flags to be used by the other DMS layers:

1. The intruder at Nine o'clock Flag (NF): When the intruder become at at 9 o'clock position relative to the UAV (usually pilots use a clock position to give the relative direction of an object). In some conflict avoidance scenarios (e.g., approaching conflict scenarios) in manned aircraft the pilot observes the intruder position while performing the avoidance manoeuvre, and when the intruder reaches the 9 o'clock position the pilot tries to restore the initial heading angle of the aircraft. Hence, the aircraft flies parallel to its previous path.
2. Collision Resolved Flag (RF): When the intruder range is greater than a predefined value with a positive relative speed V_r the collision is resolved.

4.3. Advisory System

The decision making functionality of the proposed DMS architecture is taken place in Layer-3, which receives information from Layer-1 and Layer-2, then uses the rules of the air to generate suitable Resolution Advisories (RA). The pilot or remote pilot can then uses these RA to avoid the conflict. In this paper the level flight scenarios are discussed so the horizontal advisories for avoidance manoeuvres are generated which are:

- Turn right manoeuvre: If $AF1 = 1$ and $AF2 = 0$ then the RA is single (Right), but if ($AF2 = 1$ or $CF = 1$) then the RA command is double (Right, Right) indicating that the pilot should make a greater right turn than the single (Right) command.
- Turn left manoeuvre: Same as turn right case so if $AF1 = 1$ then (Left); if $AF2 = 1$ or $CF = 1$ then (Left, Left).
- Holding the current speed and altitude: The RA in this case is (Hold).
- Level off.

Figure 6 shows the flowchart for the advisory algorithm that is used in head-on collision scenarios. It can be seen that when ($AF2$) is not activated (which means that the intruder is far enough away to make vertical manoeuvre) if the aircraft is climbing or descending the RA command is to level off the aircraft. But if the aircraft is in level flight then the RA is (Right) in case there is no offset, and (Left) in the offset case. However, if ($AF2$ or CF) are activated then the vertical resolution advisories will not be taken into consideration for safety reasons due to the aircraft inertia in vertical manoeuvre. So the advisory system will just generate the horizontal resolution advisories which are (Right, Right) when no offset exists and (Left, Left) in the case of offset.

Figure 7 gives the flowchart for the advisory algorithm that generates the resolution advisories in (overtaking, and being overtaken) scenarios. The conflict resolution advisories can be passed to the remote pilot of the UAV, or can be shown on the avionics for manned aircraft.

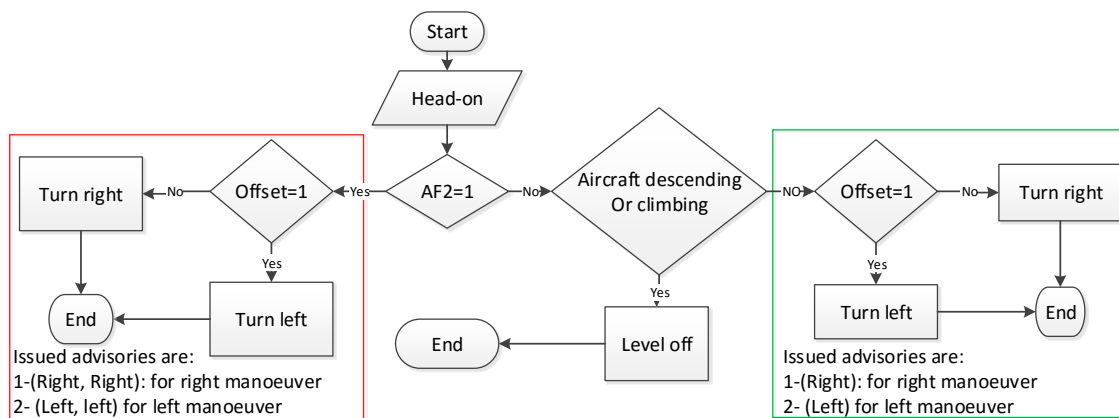


Figure 6. Flowchart for advisory generating during head-on collision.

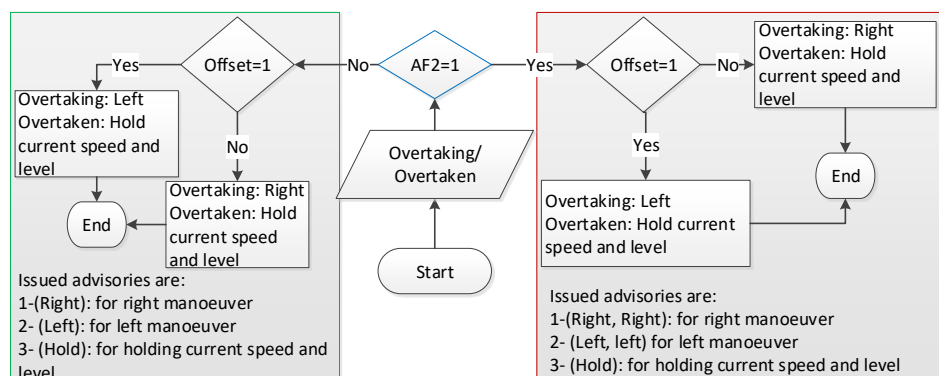


Figure 7. Flowchart for Resolution Advisory (RA) generation for overtaking/overtaken conflict scenarios.

4.4. Collision Avoidance Manoeuvre Generation

The avoidance manoeuvre generation system generates the trajectory profiles of the avoidance manoeuvre based on the information that received from the upper layers of the DMS and the rules of

the air. In order to achieve this function, a set of predefined manoeuvres are used. The parameters of the predefined manoeuvre are governed by the initial states of the UAV and the intruder, and UAV dynamics constraints. This paper proposes a combination of level flight and a coordinated turn manoeuvre with constant speed and altitude for the horizontal avoidance manoeuvres. The method here is to construct a heading rate signal for the full avoidance manoeuvre, then apply this signal to the UAV dynamic model to generate the avoidance manoeuvre trajectory profiles. The generated avoidance manoeuvre trajectory profiles are, then, either passed to the inverse dynamics to generate the UAV control inputs, or used as global trajectory profiles for the Local Trajectory Planning (LTP) algorithm that will be discussed later in this paper.

5. Avoidance Manoeuvre Trajectory Profile Generation

The avoidance manoeuvre trajectory profiles are generated as follows:

1. Specify the avoidance manoeuvre type and find its characteristics: a coordinated turn and a level flight manoeuvres are selected to perform the turning part and the straight flight part of the avoidance manoeuvre respectively.
2. Find the heading rate for each part of the avoidance manoeuvre: the coordinated turn heading rate can be found using the UAV current states and dynamic constraints. The level flight has a zero heading rate.
3. Calculate the time periods that are associated with the defined heading rates depending on the conflict scenario. Hence, the heading rate signal for the avoidance manoeuvre can be constructed.
4. Generate the avoidance manoeuvre profiles by applying the constructed heading rate to the UAV dynamic model.

5.1. Avoidance Manoeuvres' Types and Characteristics

A coordinated turn with a constant speed and altitude maneuver is preferred by pilots for the turning part of the avoidance manoeuvre [55]. Figure 8 shows the forces acting on the aircraft when performing a coordinated turn at a constant speed V .

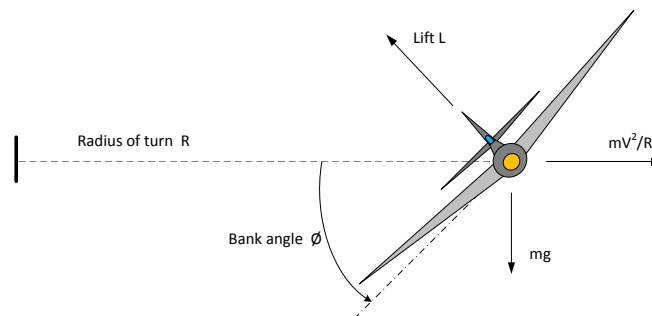


Figure 8. Forces balance in equilibrium state of turning aircraft.

The acting forces on the aircraft are balanced at an equilibrium state. Hence, the projection of the lift force on the horizontal axis and the centrifugal force are equal:

$$L \sin(\phi) = \frac{mV^2}{R} \quad (1)$$

where ϕ is the bank angle, m is the aircraft mass, V is the speed, R is the radius of turn, and L is the total aircraft lift given by [56] $L = \frac{1}{2}\rho SC_L V^2$ where ρ is the air density, S is the wing area, C_L is the lift coefficient given by $C_L = C_{L_0} + \alpha C_{L_\alpha}$ where C_{L_0} , C_{L_α} are the zero-angle-of-attack lift coefficients and α is the lift curve slope. The aircraft weight is equal to the projected lift force,

$$L \cos(\phi) = mg. \quad (2)$$

Hence, the turn radius can be calculated from (1) and (2):

$$R = \frac{V^2}{g \tan(\phi)}. \quad (3)$$

The full turn time is $t = 2\pi R/V = 2\pi V/(g \tan(\phi))$.

The heading rate $\dot{\psi}$ is given by $\dot{\psi} = 2\pi/t = g \tan(\phi)/V$. The maximum heading rate at a specific speed is

$$\dot{\psi}_{\max} = g \frac{\tan(\phi_{\max})}{V} \quad (4)$$

where ϕ_{\max} is the maximum roll angle. Constant altitude and speed during a turn is achieved by increasing the lift and thrust to compensate the vertical lift reduction and the increase in the drag. The normal load factor n is

$$n = \frac{L}{mg}. \quad (5)$$

From (2), and (5) the normal load factor during the turn can be rewritten as $n = 1/\cos(\phi)$. So the bank angle is constrained by the maximum normal load factor, $n_{\max} = \frac{1}{2}\rho V^2 S C_{L_{\max}}/(mg) = 1/\cos(\phi_{\max})$.

The stall speed increases during the turn, hence, it must be taken into consideration:

$$V_{\text{stall}(\text{turn})} = V_{\text{stall}} \sqrt{n} = \frac{V_{\text{stall}}}{\sqrt{\cos(\phi)}} \quad (6)$$

where $V_{\text{stall}(\text{turn})}$ is the turn stall speed, V_{stall} is the level flight stall speed. Equation (6) shows that the turn stall speed is increased when the the bank angle is increased. Also, the load factor must be increased to maintain the aircraft altitude.

5.1.1. Avoidance Manoeuvre Generation for Head-on/Overtaking Conflict Scenarios

The UAV performs (right/left) manoeuvres to avoid a conflict in head-on and overtaking conflict scenarios. The upper layers in the DMS determine the direction of turn. The same avoidance manoeuvre is proposed for the head-on and the overtaking conflict scenarios. A head-on avoidance manoeuvre is shown in Figure 9. The avoidance manoeuvre is divided into the following time periods:

1. Time period (T_1): perform a coordinated turn to turn right with constant heading rate $\dot{\psi}_1$ to change the heading angle by a predefined value $\Delta\psi_{T_1}$.
2. Time Period (T_2): fly straight and level with a constant speed V to achieve a predefined clearance distance d_c .
3. Time Period (T_3): a coordinated turn is initiated to turn left and so head parallel to the global trajectory. This is achieved by using a constant heading rate with the same turn rate $\dot{\psi}_3 = -\dot{\psi}_1$ to achieve heading angle change $\Delta\psi_{T_3} = -\Delta\psi_{T_1}$.
4. Time Period (T_4): fly straight and level parallel to the global trajectory until the collision assessment layer sets the resolution flag RF.

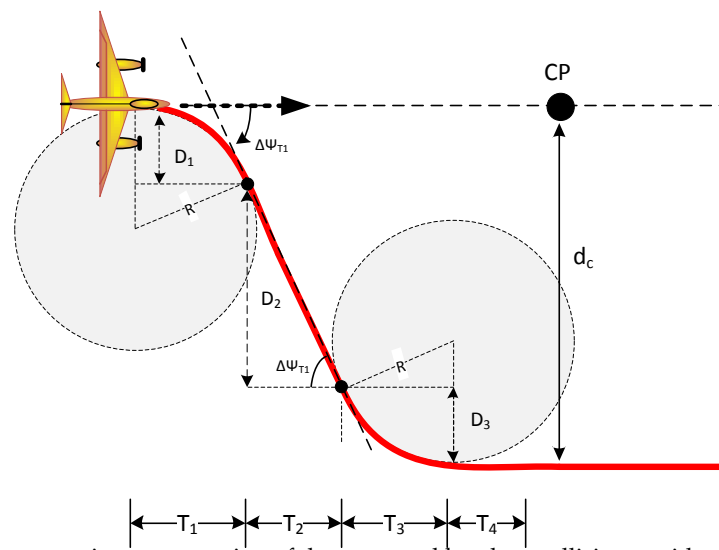


Figure 9. The geometric representation of the proposed head-on collision avoidance manoeuvre.

5.1.2. Avoidance Manoeuvre Generation for Approaching Scenarios

The rules of the air direct that the UAV must give the way to traffic that is approaching from the right side by turning right and tracking behind the approaching intruder [52]. Three types of avoidance manoeuvre are proposed using the coordinated turn with constant speed and altitude:

1. **Right-Straight-Left (RSL) manoeuvre.** This is similar to the head-on conflict avoidance manoeuvre but with a heading angle change of $\frac{\pi}{2}$ rad. However, the straight part is controlled by activation of the 9 o'clock flag NF. So the UAV turns right by $\frac{\pi}{2}$ rad, then travels straight until the intruder is at 9 o'clock relative to the UAV, at this point the UAV turns left by $\frac{\pi}{2}$ rad so the UAV is heading parallel to the global trajectory.
2. **Right-Straight then Left-Straight (RS-LS) manoeuvre.** The RSL uses the initial state of the intruder (speed and heading angle) and assumes these will be held during the conflict avoidance manoeuvre (according to the rules of the air). However, the available intruder state (measured by the on-board sensing unit, or provided by the ground station) can be mismatched with the actual one. In addition, using the intruder's initial state values to calculate the whole avoidance manoeuvre is not sufficient if these values change during the avoidance manoeuvre. Therefore, updated values of the intruder's state can be useful to reduce the mismatch effects and to overcome the intruder's state values changes problem. Hence, the RSL avoidance manoeuvre is modified to the RS-LS avoidance manoeuvre. The RS-LS avoidance manoeuvre is divided into two parts: Right-Straight (RS) that is initiated by collision flag CF, and Left-Straight (LS) that is initiated by the nine o'clock flag NF flag.
3. **Circle Manoeuvre.** A coordinated turn is used to turn right and complete a full circle. The UAV will return to the global trajectory at the end of circulation and can continue tracking the global trajectory.

Some case studies of possible right approaching conflicts are given in Figure 10 to clarify the proposed avoidance manoeuvres:

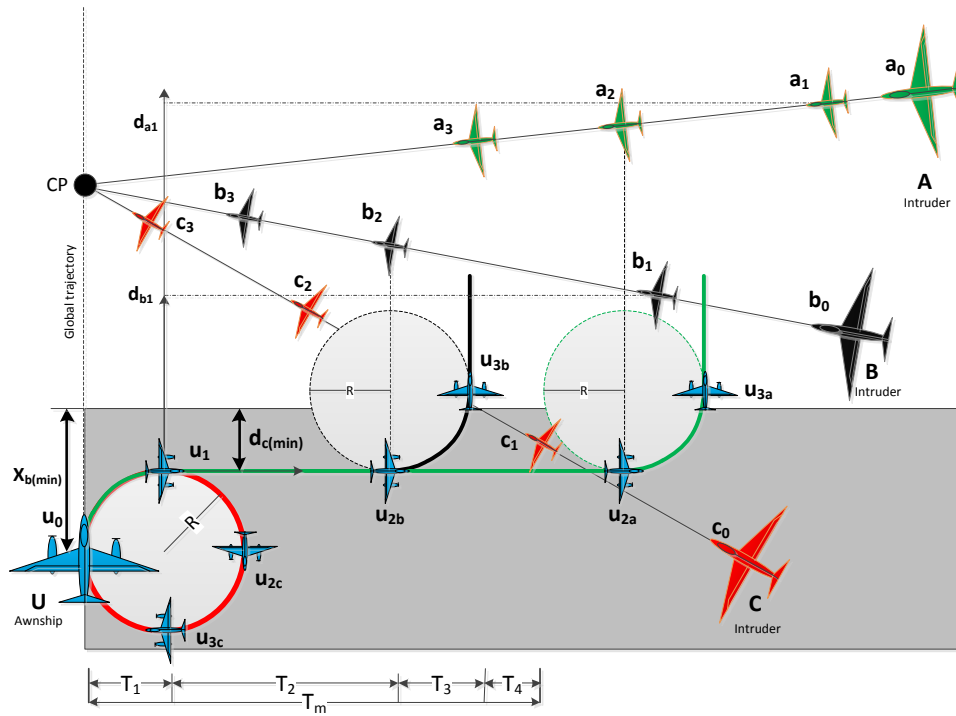


Figure 10. Avoidance manoeuvres for different right approaching conflict scenarios.

- Scenario A. The UAV and the intruder initial positions are u_0 , and a_0 respectively. The avoidance manoeuvre can be divided into four parts: time T_m is divided into four time periods:
 1. The UAV makes a full right turns manoeuvre $\Delta\psi = \frac{\pi}{2}$ rad moving from u_0 to u_1 . The time period for this part is T_1 . At the end of T_1 the intruder position is a_1 .
 2. The UAV travels straight from u_1 to u_{2a} . This part is performed during the time period T_2 . At the end of this period the intruder position is a_2 where the bearing angle is $\frac{\pi}{2}$ rad (the intruder is at 9 o'clock of UAV).
 3. The UAV turns left by $\frac{\pi}{2}$ rad during the time period T_3 moving to u_{3a} position, where, it becomes parallel to the initial path. The intruder position will be a_3 at the end of this period.
 4. The UAV then travel straight until the conflict resolution flag RF is activated where it can resume tracking the global trajectory. The time period of this part is T_4 .
- Scenario B. In this scenario the UAV and the intruder are initially at u_0 , and b_0 respectively. The avoidance manoeuvre parts for this scenario are as same as the avoidance manoeuvre parts for scenario A with shorter T_2 time period (the time periods that shown at the bottom of Figure 10 are for scenario B). The RSL or RS-LS avoidance manoeuvres are suitable for this kind of conflict. However, there are some right approaching conflicts that cannot be avoided using the RSL or RD-LS. For example, when the intruder position at the end of the time period T_1 is on the right of the UAV, or if it is at a head-on position such as the intruder in conflict scenario C. The circle avoidance manoeuvre is proposed to avoid these scenarios.
- Scenario C. The initial position of the UAV and the intruder are u_0 and c_0 respectively. At the end of time period T_1 the UAV and the intruder will be at u_1 and c_1 respectively, which means they are nearly at a head-on conflict position (as can be seen in Figure 10). The proposed manoeuvre to avoid this conflict is to make a full circle manoeuvre rather than going straight. A minimum distance $d_{c(min)}$ is used to differentiate between the intruders that need to be avoided by using the RSL/RS-LS, or the circle type avoidance manoeuvre. The RSL/RS-LS avoidance manoeuvres are used for intruders which will be out of the shaded area, that is shown in Figure 10, at the end of

T_1 time period. The circle type avoidance manoeuvre is used for the intruders that will be inside the shaded area at the end of time period T_1 . The shaded area is determined by the predefined minimum clearance distance $d_{c(\min)}$.

5.2. Avoidance Manoeuvres Parameterization

A geometric approach is proposed to parameterize the avoidance manoeuvres. The following subsection presents the proposed methodology for each conflict scenarios.

5.2.1. Head-on Manoeuvre Parameterization

The avoidance manoeuvre shown in Figure 9 can be generated by using the heading rate shown in Figure 11 as the command signal and holding the speed and altitude constant. Heading rate integration gives the heading angle for the selected manoeuvre as shown in Figure 12.

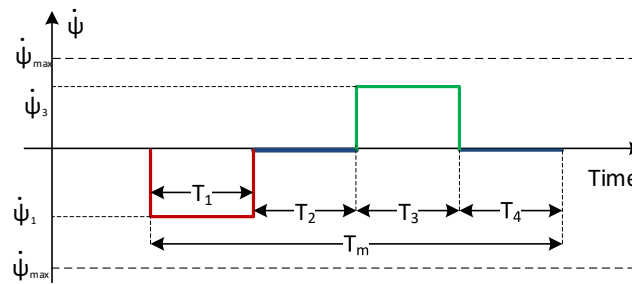


Figure 11. The heading rate of the proposed head-on collision avoidance manoeuvre.

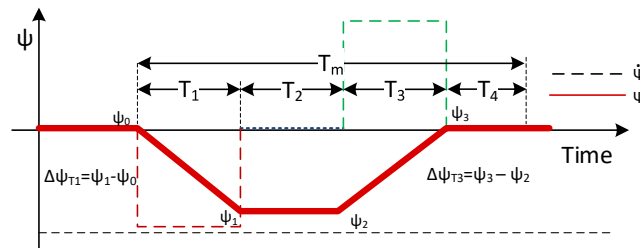


Figure 12. The heading angle of the proposed head-on collision avoidance manoeuvre.

The proposed avoidance manoeuvre can be defined by its heading rates ($\dot{\psi}_1, \dot{\psi}_3$), and the total heading angle changes ($\Delta\psi_1, \Delta\psi_3$). If these quantities are specified then the time periods (T_m, T_1, T_2, T_3, T_4) can be calculated. The heading rate and the change in heading angles can be predefined to satisfy some conditions:

1. The UAV must not head backwards during or after resolving the conflict. Thus, the heading difference is constrained so $\Delta\psi_{T_1} < \frac{\pi}{2}$. This also guarantees that the intruder will be in the Field of Regard (FOR) while the avoidance manoeuvre is performed (the requirement for the onboard sensor system for a UAV is to cover the FOR of $(\pm 110^\circ)$ horizontal with respect to the longitudinal axis of the UAV [44]).
2. The UAV must be in parallel with its initial state (i.e., $\psi_0 = \psi_3$). This can be achieved by $\Delta\psi_{T_3} = -\Delta\psi_{T_1}$ where $\Delta\psi_{T_1} = T_1\dot{\psi}_1$, and $\Delta\psi_{T_3} = T_3\dot{\psi}_3$ are the total heading changes during the time periods T_1 , and T_3 respectively so $T_3\dot{\psi}_3 = -T_1\dot{\psi}_1 \Rightarrow \dot{\psi}_1/\dot{\psi}_3 = -T_3/T_1$. In this paper $\dot{\psi}_3 = -\dot{\psi}_1$, so $T_1 = T_3$.
3. The UAV should achieve a certain clearance distance from the expected Collision Point (CP), i.e., the clearance distance should be greater than or equal to a predefined value d_c as can be seen in Figure 9. The clearance distance is the sum of the distances (D_1, D_2, D_3) that are achieved during the time periods (T_1, T_2, T_3) respectively, i.e., $D_1 + D_2 + D_3 = d_c$. This condition can be

used to find T_2 . Now $\dot{\psi}_1 = -\dot{\psi}_3$, so $D_1 = D_3$, and from geometry $D_1 = R(1 - \cos(\Delta\psi_{T_1}))$ where R is the turn radius given by (3). Hence $D_2 = d_c - 2R(1 - \cos(\Delta\psi_{T_1}))$. The relationship between the time period T_2 and the distance D_2 , at a constant speed V is

$$T_2 = \frac{D_2}{V \sin(\Delta\psi_{T_1})} = \frac{d_c - 2R(1 - \cos(\Delta\psi_{T_1}))}{V \sin(\Delta\psi_{T_1})} \quad (7)$$

where $0 < \Delta\psi_{T_1} < \frac{\pi}{2}$.

4. Calculation of T_4 , and T_m : the manoeuvre time T_m is going to be used in the next steps, for instance, in the trajectory profiles discretization, and in the speed profiles integration for position profile generation. Hence, it must be predefined as a constant value. The constant value of T_m must be enough to perform the avoidance manoeuvre and resolve the collision. The time period T_4 can be calculated as $T_4 = T_m - (T_1 + T_2 + T_3)$.

In manned aircraft, pilots use exaggerated turn manoeuvres to make other pilots aware of the collision risk. Hence, in this paper the value of the heading and the heading rate are selected to perform either an average manoeuvre ($\Delta\psi = \frac{\pi}{4}$ and $\dot{\psi}_o = [50\% - 75\%]\psi_{\max}$) or an exaggerated manoeuvre ($\Delta\psi = \frac{\pi}{3}$ and $\dot{\psi}_o = [75\% - 100\%]\psi_{\max}$). The average manoeuvre is selected when the time to collision T_c is within a predefined range value (e.g., greater than 20 s). In critical situations when the time to collision is small, the exaggerated manoeuvre is selected.

5.2.2. RSL Avoidance Manoeuvre Parameterization

The RSL avoidance manoeuvre is divided into four time periods, T_1 , T_2 , T_3 , and T_4 , as can be seen in Figure 13. The turn parts of this manoeuvre are achieved by a coordinated turn (T_1 and T_3 periods) and constant speed level flight for the straight parts. The difference between the RSL manoeuvre and the head-on manoeuvre (shown in Figure 9) is that in the RSL manoeuvre the UAV must complete the right turn, but this is not necessary for the head-on manoeuvre. Hence, the RSL has a similar shape of heading rate and heading angle those shown in Figures 11 and 12. However, the manoeuvre parameters calculation is different because $\Delta\psi_{T_1} = \Delta\psi_{T_3} = \frac{\pi}{2}$ in the RSL manoeuvre. The following steps are used to calculate the RSL avoidance manoeuvre parameters T_m , T_1 , T_2 , T_3 , and T_4 :

- Define the heading rate of turn, that can be linked to the maximum heading rate by defining two types of manoeuvre (as for the head-on/overtaking avoidance manoeuvre).
- Calculate the time period T_1 by $T_1 = \Delta\psi_{T_1} / \dot{\psi}_1 = \pi / (2\dot{\psi}_1)$.
- Calculate the time period T_2 . At the beginning of time period T_2 the UAV and intruder are at u_1 and b_1 positions respectively. They move to positions u_2 , and b_2 by the end of T_2 . Hence, the time period T_2 is given by

$$T_2 = \frac{d_{ub2}}{V + V_b \cos(\theta_b)} \quad (8)$$

where V_b is the intruder speed, θ_b is the angle between the intruder speed vector and the y -axis, and d_{ub2} is the projection of the distance between the UAV and the intruder at the end of the time period T_1 on the y -axis given by $d_{ub2} = R_{y0} - (R + d_{b1})$ where R_{y0} is the initial intruder range projection on the y -axis, R is the turn radius given by (3), and d_{b1} is the projection on the y -axis of the traveled distance by the intruder during the time period T_1 given by $d_{b1} = V_b \cos(\theta_b) T_1$. Unlike the avoidance manoeuvre for the head-on/overtaking conflict scenarios, the RSL avoidance manoeuvre calculation depends on the intruder speed and heading (during the time period T_2).

- Time period T_3 is equal to T_1 so that the UAV heading angle at the end of T_3 is the same as the UAV initial heading angle ($\dot{\psi}_3 = -\dot{\psi}_1$).
- Time period T_4 is calculated by $T_4 = T_m - (T_1 + T_2 + T_3)$.

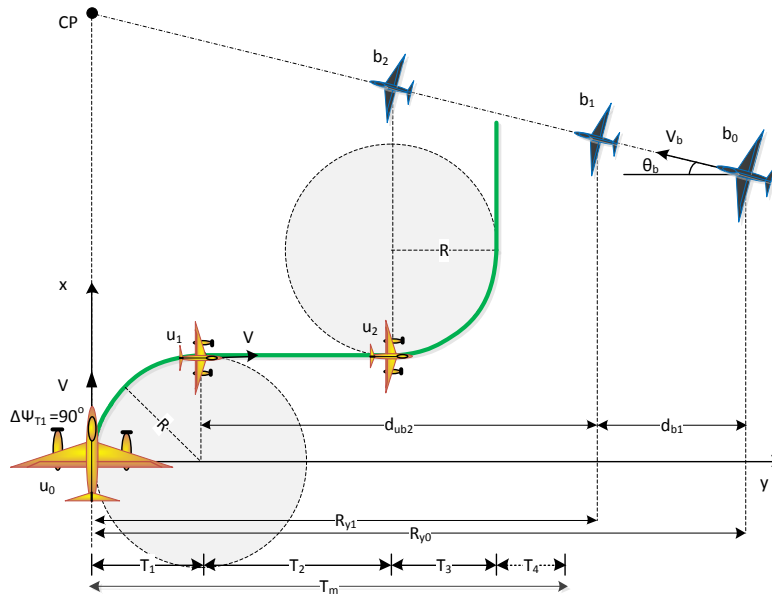


Figure 13. The Right-Straight-Left (RSL) avoidance manoeuvre for right approaching conflict scenario.

5.2.3. RS-LS Avoidance Manoeuvre Parameterization

The RS-LS avoidance manoeuvre is divided into two parts: Right-Straight RS that is initiated by the CF flag, and Left-Straight LS that is activated by the NF flag. Figure 14 illustrates the sequence of the RS-LS avoidance manoeuvre parts with the collision assessment flags; the heading rate values during the time periods are shown at the bottom of the figure. The time values T_{m1} , and T_{m2} are predefined to be large enough to perform the avoidance manoeuvre. Time periods T_1 , and T_3 are calculated the same as for the RSL avoidance manoeuvre. The time period $T_2 = T_{m1} - T_1$ is interrupted by the NF flag, and the time period $T_4 = T_{m2} - T_3$ is interrupted by the RF flag.

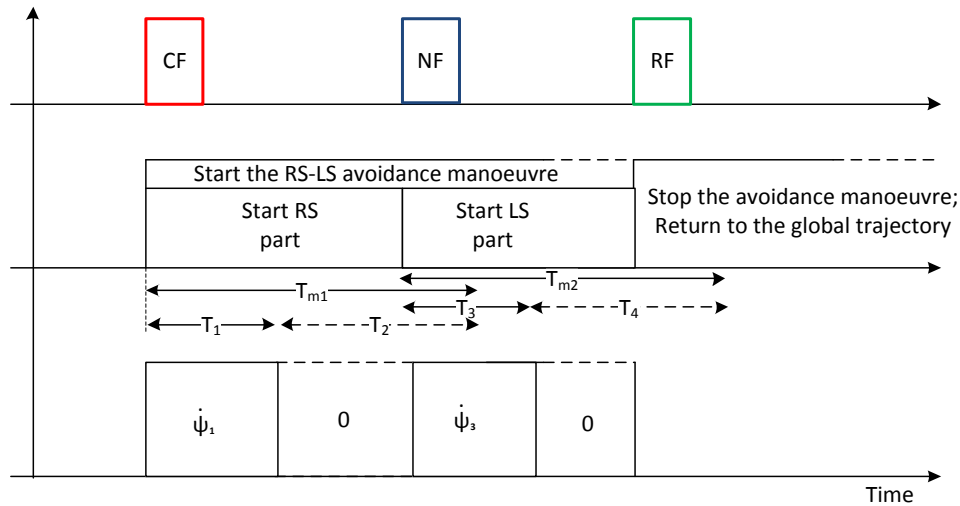


Figure 14. The Right-Straight then Left-Straight (RS-LS) avoidance manoeuvre parts sequence.

5.2.4. Circle Avoidance Manoeuvre Parameterization

A geometric presentation of the circle avoidance manoeuvre is shown in Figure 15. The condition to perform the circle manoeuvre instead of the RSL/RS-LS avoidance manoeuvre is:

$$R_{x1} \leq X_{\min} \Rightarrow R_{x1} \leq R + d_{c(\min)} \quad (9)$$

where R_{x1} is the intruder range projection on the x -axis, when it at the position c_1 (the intruder ranges are measured from the reference frame origin at u_0), R is the turn radius, and $d_{c(\min)}$ is a predefined minimum clearance distance, measured from the UAV position u_1 (when it has completed a $\frac{\pi}{2}$ rad right turn). The intruder range projection R_{x1} is given by $R_{x1} = R_{x0} + d_{c1}$, where R_{x0} is the intruder range projection on the x -axis when it is at the position c_0 , and d_{c1} is the projection of the intruder traveled distance from c_0 to c_1 given by $d_{c1} = \frac{T_t}{4} V_c \sin(\theta_c)$, where V_c is the intruder speed, θ_c is the angle between the y -axis and the intruder speed vector as shown in Figure 15, and T_t is a full circle turn time given below by (11). Then the circle avoidance manoeuvre condition can be written:

$$R_{x0} \leq R + d_{c(\min)} - \frac{T_t}{4} V_c \sin(\theta_c). \quad (10)$$

This avoidance manoeuvre is simply a circle with a constant speed and altitude (coordinated turn). The circle radius is controlled by the selected heading rate $\dot{\psi}$ shown in Figure 16, and the UAV speed V since the radius is given by (3). The UAV heading angle is shown in Figure 17. As can be seen in Figures 16 and 17, the manoeuvre is divided into two periods:

- the circle turn time period T_t :

$$T_t = \frac{\Delta\psi_{T_t}}{\dot{\psi}_1} = \frac{2\pi}{\dot{\psi}_1} \quad (11)$$

where $\dot{\psi}_1$ is the selected avoidance heading rate and

- the time period $T_4 = T_m - T_t$ where the manoeuvre time T_m is predefined.

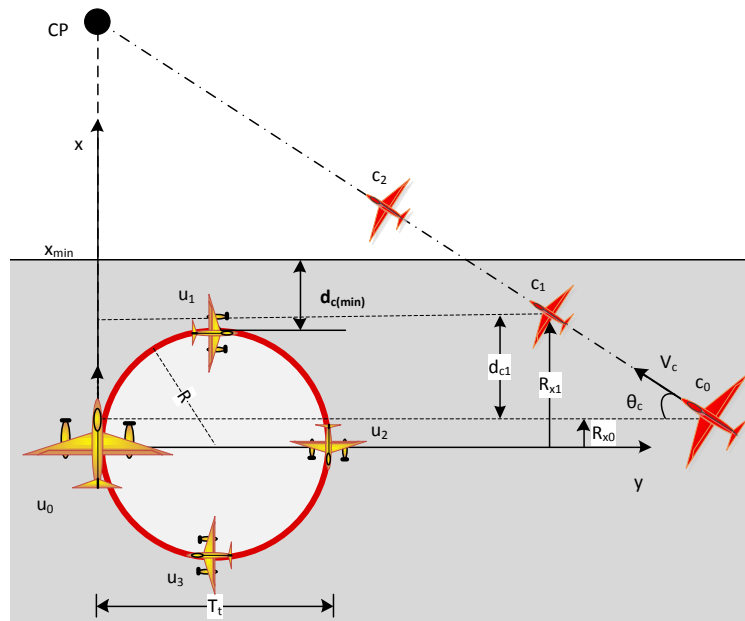


Figure 15. The geometric representation of the proposed circle avoidance manoeuvre.

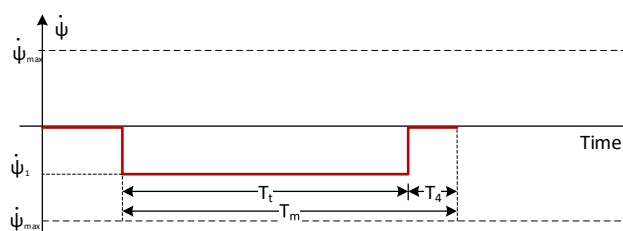


Figure 16. Heading rate for the circle avoidance manoeuvre.

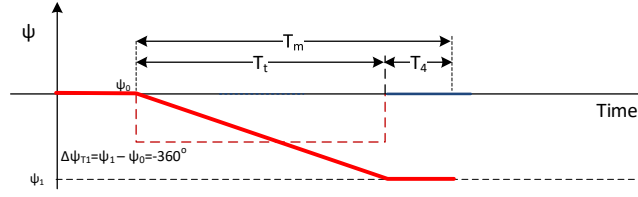


Figure 17. Heading angle for the circle avoidance manoeuvre.

5.3. Trajectory Profiles Generation and Parameterization

Trajectory profiles are required for the next steps of the collision avoidance system. The constructed signal of the heading rate of the avoidance manoeuvre is used as the input for the UAV lateral directional model to generate the avoidance manoeuvre trajectory profiles (speed, and position). The lateral directional dynamics of the UAV under constant speed and altitude is given by [57]:

$$\begin{aligned}\dot{x} &= V \cos \psi \\ \dot{y} &= V \sin \psi \\ \dot{\psi} &= U\end{aligned}\quad (12)$$

the pair (x, y) represents the UAV position, V is the UAV speed, and U is the input signal. A discrete representation of this model is formed by an Euler approximation:

$$\begin{aligned}x(k+1) &= x(k) + T_s V \cos \psi(k) \\ y(k+1) &= y(k) + T_s V \sin \psi(k) \\ \psi(k+1) &= \psi(k) + T_s U(k)\end{aligned}\quad (13)$$

where T_s is the sampling time and $k = 0, 1, \dots, n$ are the discrete steps. The relationship between the time manoeuvre and sampling time is $n = \frac{T_m}{T_s}$. Hence, the constructed heading rate signal is discretized then used to generate the discrete speed and position profiles of the proposed avoidance manoeuvre,

$$\psi = [\psi(0), \psi(1), \dots, \psi(n-1), \psi(n)] \quad (14)$$

The generated trajectory profiles (position, and speed) can then be curve-fitted to calculate all the trajectory profiles (position, speed, acceleration, and the rate of acceleration or jerk) which can be used as a global trajectory for the LTP algorithm. The inverse dynamic algorithm that generates the command signal for the UAV depends on the trajectory profiles (position, speed, acceleration, and rate of acceleration) in the 3D reference frame. However, the discretized lateral directional model of the UAV (13) with the constructed heading rate signal as input generates only the speed and position profiles in the 2D reference frame. Thus NURBS curves are used to describe the trajectory profiles. Appendix A gives an introduction to NURBS curves and their properties.

5.3.1. Avoidance Trajectory Parametrization

A 6th order Bezier curve [58] is used to curve fit the generated speed profiles. Using (A2)–(A4), the 6th order Bezier curve basis functions are

$$\begin{aligned}B_0 &= (1 - \tau)^6, & B_1 &= 6\tau(1 - \tau)^5, & B_2 &= 15\tau^2(1 - \tau)^4, \\ B_3 &= 20\tau^3(1 - \tau)^3, & B_4 &= 15\tau^4(1 - \tau)^2, & B_5 &= 6\tau^5(1 - \tau), & B_6 &= \tau^6\end{aligned}\quad (15)$$

Then the Bezier curve coefficients are used to generate the avoidance manoeuvre trajectory profiles in the 3D reference frame. The curve fitting algorithm uses the least squares curve fitting technique. The coefficients of the Bezier curve that fit the speed profiles are calculated as:

$$\mathbf{C}^{\dot{u}} = \mathbf{B}_{ls} \dot{x}, \quad \mathbf{C}^{\dot{v}} = \mathbf{B}_{ls} \dot{y}, \quad \mathbf{C}^{\dot{w}} = \mathbf{B}_{ls} \dot{z} = 0 \quad (16)$$

where $\mathbf{C} = [c_0, c_1, \dots, c_6]$ are the Bezier curve coefficients, \mathbf{B}_{ls} is the curve-fit matrix which is calculated off-line using the Bezier basis function matrices $\mathbf{B}_{ls} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T$. Then the calculated coefficients of the Beizer curve are used to calculate the trajectory profiles (position, speed, acceleration, and rate of acceleration) in the 3D reference frame. The curve fitted speed profiles in forward (u), lateral (v), and vertical (w) axes are written:

$$\begin{aligned} u(\tau) &= c_0^{\dot{u}} B_0(\tau) + c_1^{\dot{u}} B_1(\tau) + c_2^{\dot{u}} B_2(\tau) + \dots + c_6^{\dot{u}} B_6(\tau) \\ v(\tau) &= c_0^{\dot{v}} B_0(\tau) + c_1^{\dot{v}} B_1(\tau) + c_2^{\dot{v}} B_2(\tau) + \dots + c_6^{\dot{v}} B_6(\tau) \\ w(\tau) &= c_0^{\dot{w}} B_0(\tau) + c_1^{\dot{w}} B_1(\tau) + c_2^{\dot{w}} B_2(\tau) + \dots + c_6^{\dot{w}} B_6(\tau) \end{aligned} \quad (17)$$

However, in horizontal manoeuvres the vertical speed is zero ($w(\tau) = 0$) hence:

$$c_i^{\dot{w}} = 0 \text{ for } i = 0, 1, \dots, 6 \quad (18)$$

The relationship between the parameter τ and the avoidance manoeuvre time (T_m) can be used to represent the time $t = \tau T_m$. Therefore, the acceleration and the jerk profiles are given by:

$$\frac{du}{dt} = \frac{1}{T_m} \left(c_0^{\ddot{u}} \frac{dB_0(\tau)}{d\tau} + \dots + c_6^{\ddot{u}} \frac{dB_6(\tau)}{d\tau} \right) \quad (19)$$

and

$$\frac{d^2u}{dt^2} = \frac{1}{T_m^2} \left(c_0^{\dddot{u}} \frac{d^2B_0(\tau)}{d\tau^2} + \dots + c_6^{\dddot{u}} \frac{d^2B_6(\tau)}{d\tau^2} \right) \quad (20)$$

The acceleration and the jerk profiles for the lateral axis can be calculated in a similar way. The acceleration and jerk profiles for the vertical axis are both zero. The position profile is driven by integration of the basis functions with respect to time t

$$\mathbf{B}_i^{int} = \int_0^{T_m} B_i(t) dt \text{ for } i = 0, 1, \dots, 6 \quad (21)$$

Hence, the trajectory profiles calculation is reduced to simple matrix multiplication:

$$u = \mathbf{C}^{\dot{u}T} \mathbf{B}, \quad \dot{u} = \frac{1}{T_m} \mathbf{C}^{\dot{u}T} \mathbf{B}', \quad \ddot{u} = \frac{1}{T_m^2} \mathbf{C}^{\dot{u}T} \mathbf{B}'', \quad x = x_0 + \mathbf{C}^{\dot{u}T} \mathbf{B}^{int} \quad (22)$$

$$v = \mathbf{C}^{\dot{v}T} \mathbf{B}, \quad \dot{v} = \frac{1}{T_m} \mathbf{C}^{\dot{v}T} \mathbf{B}', \quad \ddot{v} = \frac{1}{T_m^2} \mathbf{C}^{\dot{v}T} \mathbf{B}'', \quad y = y_0 + \mathbf{C}^{\dot{v}T} \mathbf{B}^{int} \quad (23)$$

$$w = \mathbf{C}^{\dot{w}T} \mathbf{B}, \quad \dot{w} = \frac{1}{T_m} \mathbf{C}^{\dot{w}T} \mathbf{B}', \quad \ddot{w} = \frac{1}{T_m^2} \mathbf{C}^{\dot{w}T} \mathbf{B}'', \quad z = z_0 + \mathbf{C}^{\dot{w}T} \mathbf{B}^{int} \quad (24)$$

where \mathbf{B} is the discretized basis function matrix given by:

$$\mathbf{B} = \begin{pmatrix} B_0(\tau_1) & B_0(\tau_2) & \dots & B_0(\tau_n) \\ B_1(\tau_1) & B_1(\tau_2) & \dots & B_1(\tau_n) \\ \vdots & \vdots & \dots & \vdots \\ B_6(\tau_1) & B_6(\tau_2) & \dots & B_6(\tau_n) \end{pmatrix} \quad (25)$$

\mathbf{B}' , \mathbf{B}'' are the discretized basis function derivatives matrices, and $\mathbf{C}^{\dot{u}T}$, $\mathbf{C}^{\dot{v}T}$, $\mathbf{C}^{\dot{w}T}$ are the vectors of coefficients for forward, lateral, and vertical axis:

$$\mathbf{C}^{\dot{u}^T} = \begin{bmatrix} c_0^{\dot{u}} & c_1^{\dot{u}} & \cdots & c_6^{\dot{u}} \end{bmatrix}; \quad \mathbf{C}^{\dot{v}^T} = \begin{bmatrix} c_0^{\dot{v}} & c_1^{\dot{v}} & \cdots & c_6^{\dot{v}} \end{bmatrix}; \quad \mathbf{C}^{\dot{w}^T} = \begin{bmatrix} c_0^{\dot{w}} & c_1^{\dot{w}} & \cdots & c_6^{\dot{w}} \end{bmatrix} \quad (26)$$

6. Simulation Results of the Proposed Predefined Avoidance Manoeuvres

This section discusses simulation results for some avoidance manoeuvre scenarios. MATLAB/Simulink is used to simulate the proposed avoidance manoeuvres for different collision scenarios. Figure 18 shows the block diagram that is used to generate the simulation results. The UAV with the controllers are modeled by the point-mass model, and it is assumed that the UAV is tracking the generated command signals exactly. All the required parameters of the UAV and intruders are assumed to be available for the DMS algorithm (the obstacle detection issue is not in the scope of this paper). The switch and the dashed link that are shown in the block diagram represent the method for generating the command signals by passing the trajectory profiles, that are generated by the curve fitting algorithm, directly to the inverse dynamics.

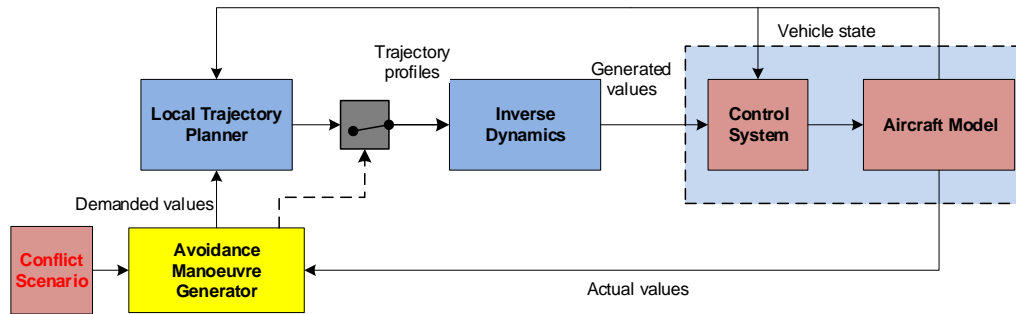


Figure 18. Block diagram for Collision Avoidance Systems (CAS) simulation.

6.1. Right Approaching Conflict Scenario Simulation Results (RSL manoeuvre)

Figure 19 shows a right approaching scenario used to test the RSL avoidance manoeuvre. The initial state for the UAV and the intruder are:

1. UAV initial state: $V = 75$ m/s, $\psi_0 = 0$ rad, and $\phi_{\max} = \frac{\pi}{3}$ rad
2. Intruder initial state: $V_b = 75$ m/s, $\psi_b = \frac{\pi}{3}$ rad, that means $\theta_b = \frac{\pi}{6}$ rad, $R_{y0} = 1500$ m, so $T_c = 20$ s

The exaggerated type of the RSL avoidance manoeuvre is selected ($\dot{\psi}_1 = \dot{\psi}_{\max} = 0.2263$ rad/s), so the turn radius is $R = V / \dot{\psi}_1 = 331$ m, the time period $T_1 = \pi / (2\dot{\psi}_1) = 7$ s, $R_{y0} = 1500 \cos(\frac{\pi}{6}) = 1300$ m, and $d_{b1} = V_b \cos(\frac{\pi}{6}) T_1 = 455$ m. Time period T_2 is calculated using (8): $T_2 = 4$ s. The manoeuvre time $T_m = 50$ s, so $T_4 = T_m - (T_1 + T_2 + T_3) = 32$ s (because $T_3 = T_1 = 7$ s).

Figure 20 shows some simulation results for this scenario. The left column subplots show the heading rate, heading angle, roll angle, and flight path angle (from top to bottom).

1. The input (heading rate) to generate the proposed RSL avoidance manoeuvre is shown in the first subplot. It can be seen that the heading rate is at the maximum value (exaggerated type avoidance manoeuvre).
2. The demanded heading angle ψ_d and the generated heading angle ψ_g are shown in the second subplot. It can be seen that the generated heading angle tracks the demanded one with some deviations.
3. The demanded and the generated roll angles, ϕ_d and ϕ_g , are shown in the third subplot. The generated roll angle ϕ_g tracks the demanded one with some differences resulting from the sudden change in the demanded value, but it does not exceed the maximum heading angle.
4. The demanded and generated flight path angles are shown in the fourth subplot. As the avoidance manoeuvre is proposed to be performed in the horizontal plane the demanded flight path angle γ_d is zero. The generated flight path angle γ_g fluctuates around zero within the range $\pm 3^\circ$, which is small enough to be considered negligible.

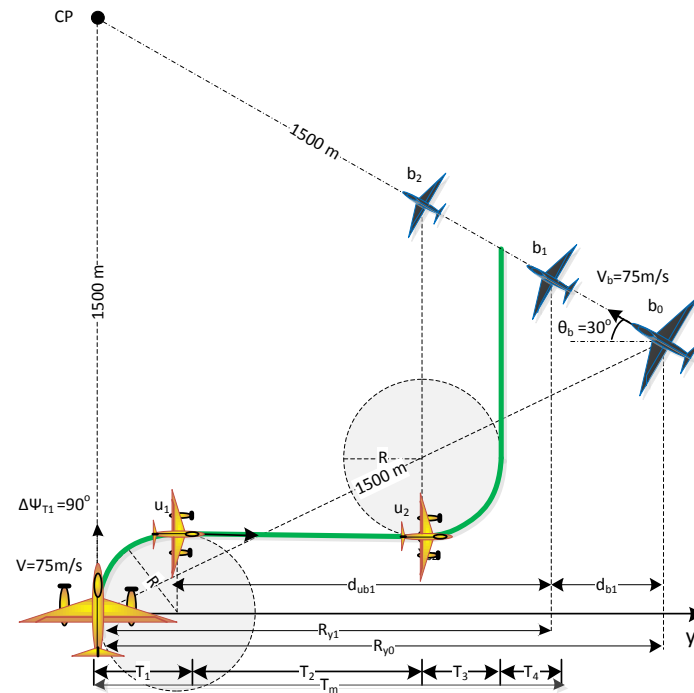


Figure 19. Right approaching scenario.

The right column of subplots in Figure 20 gives the simulation result of the speed V , forward speed u , lateral speed v , and vertical speed w . It can be seen that the generated speeds (dashed lines) are tracking the demanded speeds (solid lines). The generated UAV speed V_g is approximately 75 m/s during the manoeuvre time. Figure 21 gives the 3D representation of the manoeuvre which shows that the clearance distance is greater than the specified value that is 500 m.

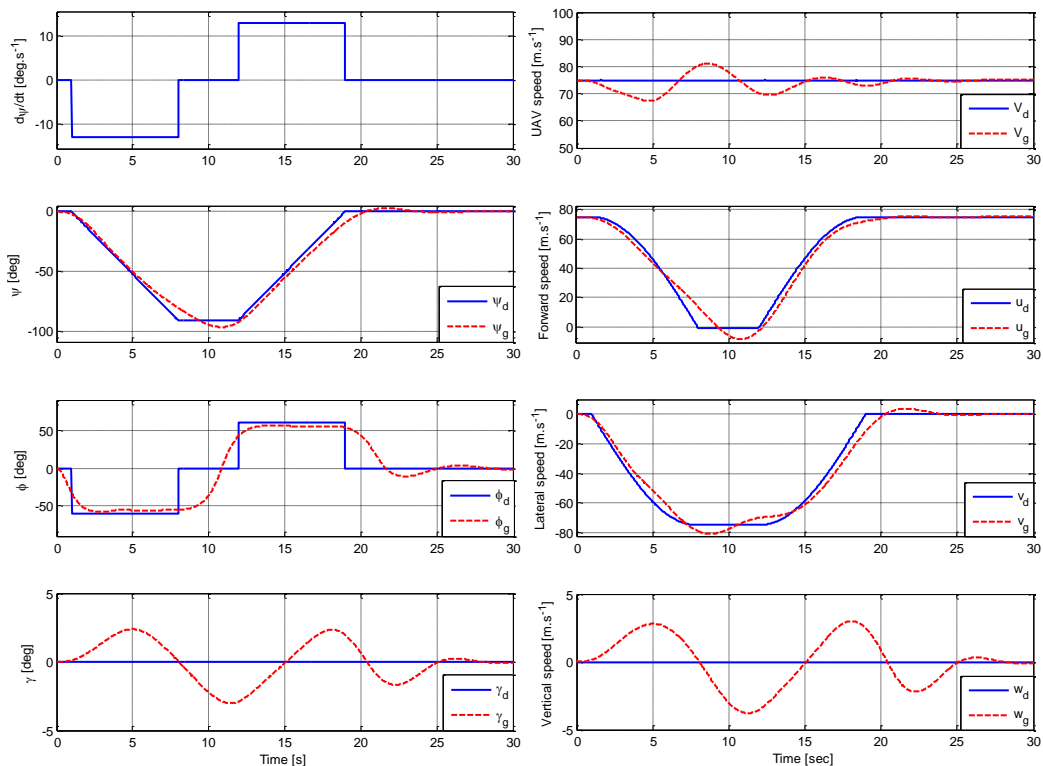


Figure 20. Simulation results of attitude (heading rate, heading angle, roll angle, and flight path angle) for the RSL avoidance manoeuvre.

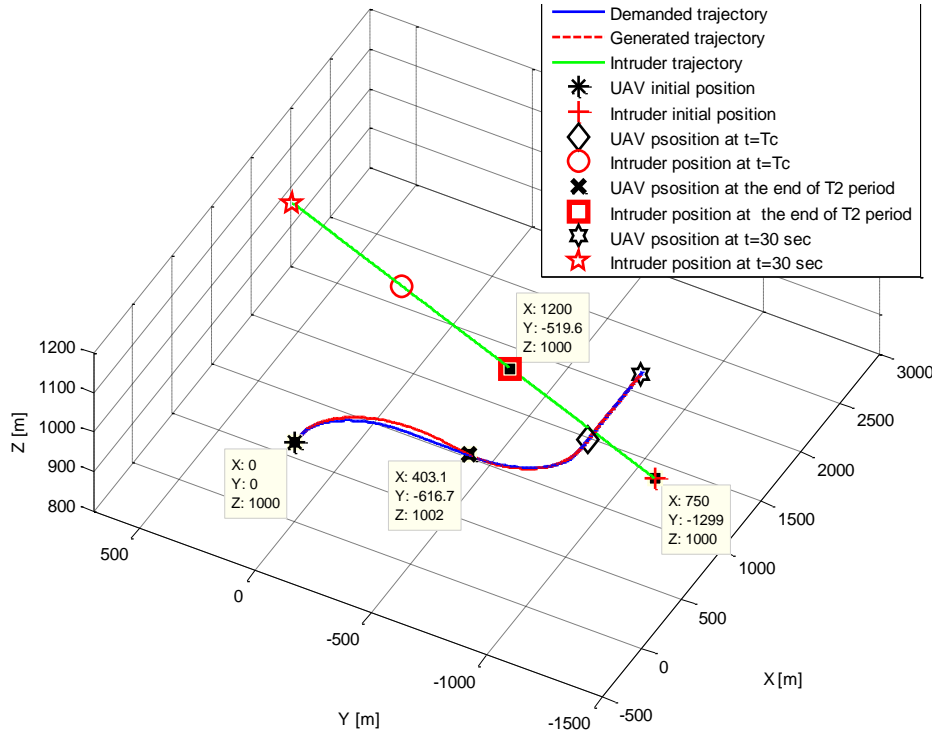


Figure 21. 3D view of the Unmanned Aerial Vehicle (UAV) trajectory for the RSL avoidance manoeuvre.

6.2. Right Approaching conflict Scenario Simulation Results (Circle manoeuvre)

Figure 22 shows a right approaching scenario used to test the circle avoidance manoeuvre. The initial state for the UAV and the intruder are:

1. UAV initial state: $V = 60$ m/s, $\psi_0 = 0$ rad, $\phi_{\max} = \frac{\pi}{3}$ rad.
2. Intruder initial state: $V_b = 75$ m/s, $\psi_c = 0.89$ rad, that means $\theta_c = 0.68$ rad, $Range = 1500$ m, and $T_c = 25$ s.

An exaggerated type of the circle avoidance manoeuvre is selected (the heading rate is $\psi_1 = 0.8\psi_{\max} = 0.2263$ rad/s), so the turn radius is $R = V/\psi_1 = 265$ m, and the time period $T_t = 2\pi/\psi_1 = 27.76$ s. The predefined manoeuvre time is selected to be $T_m = 40$ s, thus $T_4 = T_m - (T_t) = 12.24$ s, and $d_{c1} = V_c \cos(\theta_c) \frac{T_t}{4} = 455$ m.

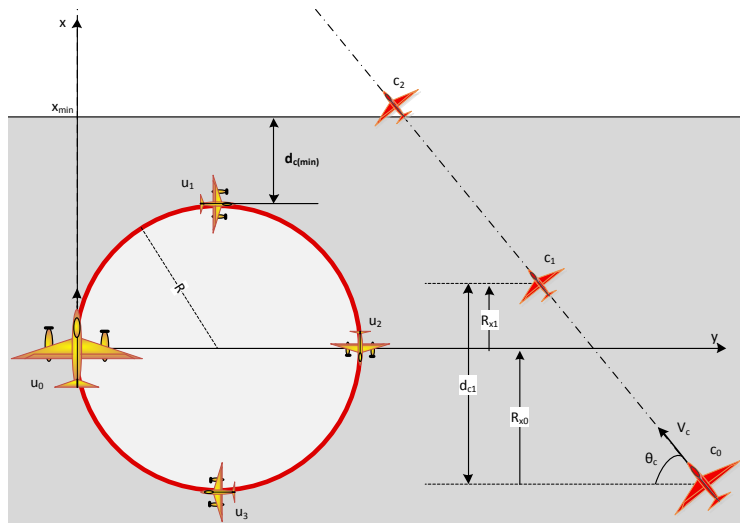


Figure 22. The circle avoidance manoeuvre for a right approaching conflict scenario.

The simulation results of this scenario are shown in Figure 23. The heading rate, heading angle, roll angle, and flight path angle are shown in left column subplots (from top to bottom). It can be seen that the generated signals track the demanded signals with acceptable fluctuations. For example, the generated heading angle tracks the demanded one with very small deviations, also the generated flight path angle tracks zero value with maximum error less than 2° . However, the UAV dynamics lags can be obviously noticed in the roll angle response in the third subplot.

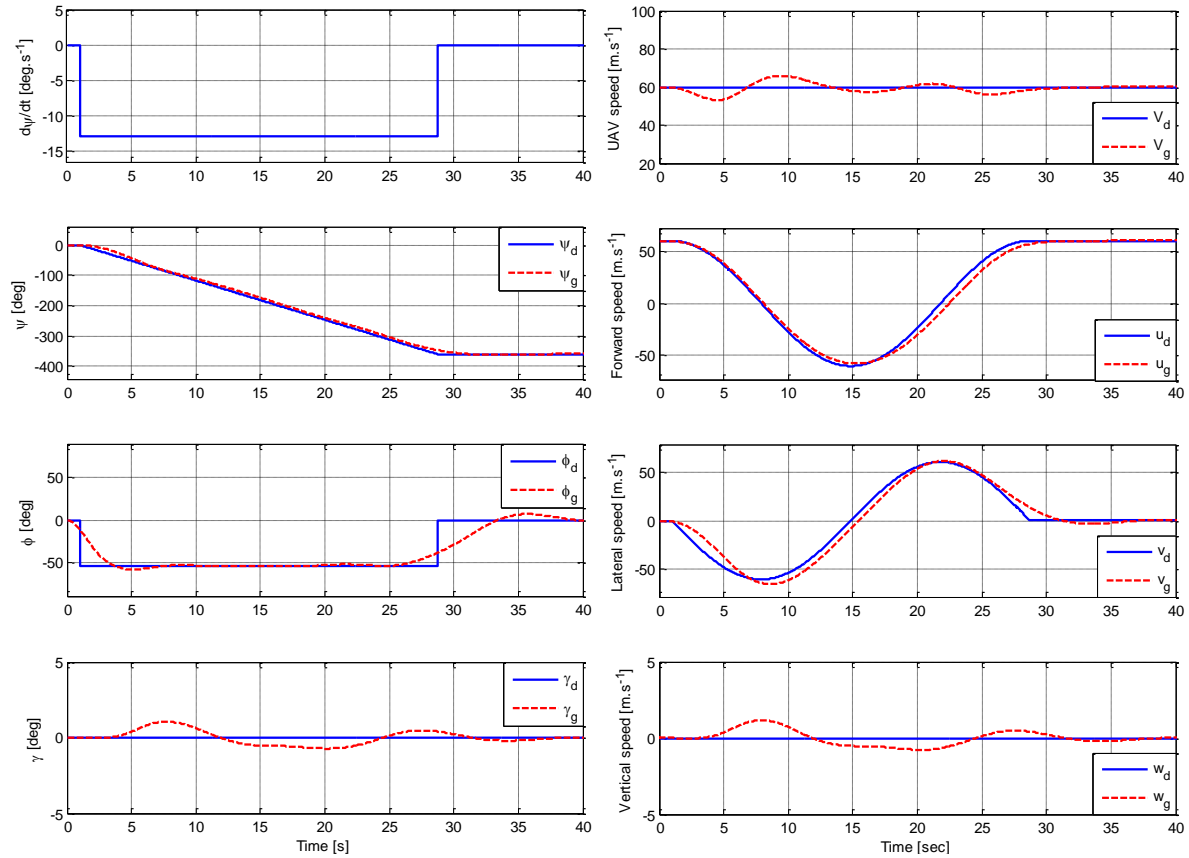


Figure 23. Simulation results of attitude (heading rate, heading angle, roll angle, and flight path angle) for the circle avoidance manoeuvre.

The simulation result of the UAV speed V , forward speed u , lateral speed v , and vertical speed w are shown in the right column of subplots. The generated speeds (dashed lines) track the demanded speeds (solid lines) with very small differences. The generated UAV speed V_g is approximately 60 m/s during the manoeuvre time. A 3D representation of the UAV avoidance manoeuvre trajectory, and the intruder trajectory are shown in Figure 24, in which the position of the UAV and the intruder at ($t = \frac{T_t}{4}$, $t = \frac{T_t}{2}$, and $t = T_m$) are highlighted to show the need for performing the circle avoidance manoeuvre rather than the RSL/RS-LS manoeuvres. At $t = \frac{T_t}{4}$ the UAV at position is (333, -213, 1003) m, and the intruder position is (154, -1090, 1000) m so the intruder is at the right of the UAV where the circle avoidance manoeuvre is the suitable option to avoid the conflict.

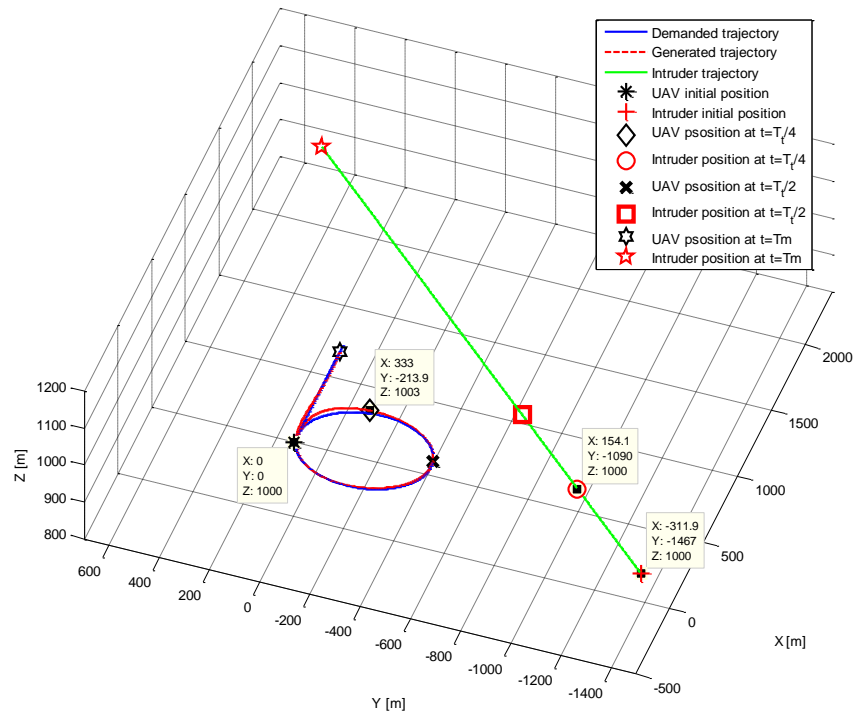


Figure 24. 3D view of the UAV trajectory for the circle avoidance manoeuvre.

7. Local Trajectory Planning

Trajectory planners can be divided into two main categories [29]: global planners which require good knowledge about the environment that the aircraft is going to fly in, and local trajectory planners which are algorithms that run continuously in order to allow the aircraft to deal with unforeseen events that may happen during the flight.

This section presents an approach for generating collision avoidance trajectories based on B-spline curves. Essentially, a finite-horizon optimal control problem is periodically solved in real-time hence updating the aircraft trajectory to avoid obstacles and drive the aircraft to its global path. The proposed approach can be summarized as follows:

1. Given a global trajectory that the aircraft is required to follow, solve the following optimal control problem:

$$\min_{\mathbf{u}(t) \in \mathcal{U}} J(\mathbf{u}(t)) \quad (27)$$

where \mathbf{u} is the control, \mathcal{U} is the feasible space of control, and J is a cost measured over a finite time horizon, $t \in [t_0, t_f]$, that drives the local trajectory to the global trajectory. Subject to the aircraft dynamics constraints pair, state constraint given by:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{u}) \quad (28)$$

where the state $\mathbf{X}(t) \in \mathcal{X}$, and aircraft trajectory obstacles constraint given by:

$$\mathbf{Y} = g(\mathbf{X}) \quad (29)$$

where the output $\mathbf{Y}(t) \in \mathcal{Y}$. Where \mathcal{X} and \mathcal{Y} are the feasible space of the state and the output respectively.

2. The problem is solved by a direct method by inverting the dynamics, so the optimization is performed in the output space $\mathbf{Y}(t) \in \mathcal{Y}$, and parameterizing the trajectory by a spline function. The cost is augmented to maintain the constraints.

3. The generated local trajectory allows the UAV to track the global trajectory while avoiding any intruder or conflict scenarios that may occur. The local trajectory optimization is periodically solved on-line in a receding horizon approach to account for system uncertainties and obstacle changes.

7.1. Differential Flatness of the Fixed-Wing Aircraft

A fixed wing aircraft dynamics can be expressed by a three Degree of Freedom (3-DoF) point-mass model [55].

Figure 25 shows the coordinate system used for the derivation of the point-mass model. It is assumed that the aircraft mass is constant and flying in still wind with zero sideslip and zero angle of attack (flight path angle γ equals the pitch attitude θ). Thus, the aircraft mathematical model is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\gamma} \\ \dot{\chi} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V \cos \gamma \cos \chi \\ V \cos \gamma \sin \chi \\ V \sin \gamma \\ (g/V)(n \cos \phi - \cos \chi) \\ (g/V)(n \sin \phi / \cos \gamma) \\ (T - D)/m - g \sin \gamma \end{bmatrix} \quad (30)$$

where x, y, z are the aircraft center of gravity coordinates in earth axis, γ is the flight-path angle, χ is the heading angle, V is the aircraft speed, g is the gravity acceleration, ϕ is the bank angle, T is the thrust, D is the drag, m is the total mass, $n = L/(mg)$ is the load factor and L is the total aircraft lift. The input and output vectors are defined respectively as:

$$\mathbf{U} = [\phi \quad T \quad n]^T, \quad \mathbf{Y} = [x \quad y \quad z]^T \quad (31)$$

The optimal control problem is formulated in output space rather than control or input space thanks to the differentially flatness property of the fixed wing aircraft model. If system's state variables and inputs can be expressed as functions of the output vector and the derivatives of the output vector the system is differentially flat [59,60]. Modifying (30) obtains:

$$V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \quad (32)$$

$$\gamma = \arcsin(\dot{z}/V) \quad (33)$$

$$\chi = \arcsin(\dot{y}/(V \cos \gamma)) \quad (34)$$

$$\phi = \arctan((\dot{\chi}V \cos \gamma)/(g \cos \gamma + V\dot{\gamma})) \quad (35)$$

$$n = (g \cos \gamma + V\dot{\gamma})/(g \cos \phi) \quad (36)$$

$$T = D + m\dot{V} + mg \sin \gamma \quad (37)$$

The aerodynamic drag is given by [56]:

$$D = \frac{1}{2}\rho S C_D V^2 \quad (38)$$

where $C_D = C_{D0} + kC_L^2$ is the drag coefficient, $C_L = 2nm g/(\rho S V^2)$ is the lift coefficient, ρ is the air density, C_{D0} is the minimum drag coefficient of the aircraft and S is the wing area. Equations (32) to (38) present the inputs and the states of the system as functions of the output vector and its derivatives, hence the system is differentially flat. So the optimal problem can be formulated in output space rather than control space. Thus, a sufficient description for the output space (trajectory profiles in our case) is needed to make the optimal problem more tractable.

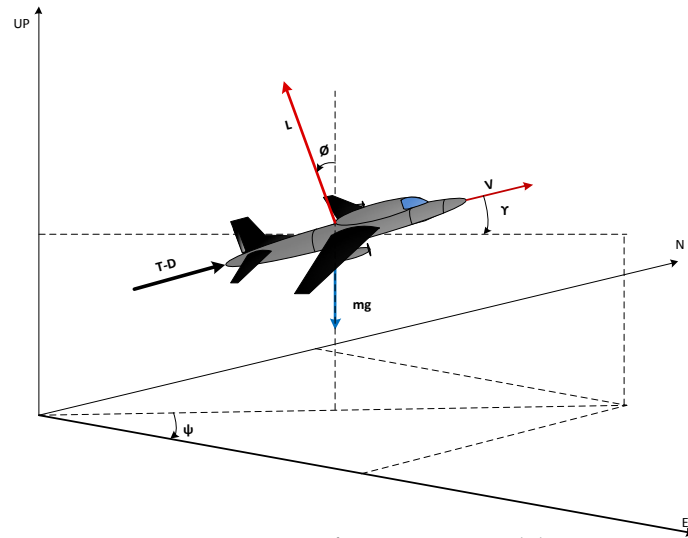


Figure 25. Aircraft point-mass model.

7.2. Local Trajectory Optimization

The optimal local trajectory profiles can be achieved by finding values of design variables that minimize a defined cost function and satisfy all constraints. Hence, the trajectory must be parameterized to find these design variables.

Trajectory Profiles Description Using B-Spline

The local trajectory profiles are described using 6th order Bezier functions, the same method as the global trajectory profiles discussed in Section 5.3.1. The selection of 6th order polynomial functions is justified in [61] where it gives good flexibility over the design horizon with an acceptable number of design variables for trajectory planning for a quad-rotor UAV. Hence, this can be applied for fixed-wing aircraft which generally have less maneuverability than quad-rotor vehicles. Equations (17)–(26) are used in a similar way to parameterize the local trajectory profiles with two differences: the relationship between the curve parameter τ and time t can be defined as $t = \tau t_h$ where a fixed time horizon (t_h) is used, and the vertical speed profile is not zero, so the term $C^{iv} = B_{is}\dot{z}$ in (16) is not zero for the local trajectory profile calculation.

There are 21 coefficients to be determined (seven for each axis) by the optimal control problem as the design variables. Using the initial boundary conditions at $\tau = 0$ gives:

$$c_0^u = u_0, \quad c_1^u = \frac{t_h}{6}\dot{u}_0 + c_0^u, \quad c_2^u = \frac{t_h^2}{30}\ddot{u}_0 - c_0^u + 2c_1^u \quad (39)$$

where u_0 , \dot{u}_0 , and \ddot{u}_0 are the initial values forward speed, acceleration and jerk respectively. Thus the first three coefficients for each trajectory profile are calculated, so the number of design variables is reduced to 12.

The computational time of the optimization problem can be reduced by converting it to unconstrained optimal control problem. This is achieved by augmenting the aircraft and the obstacles constraints in the cost function by using a penalty function method [61]. The Yukawa potential function [62] is used as the penalty function to represent the aircraft performance constraints, this ensures that the resulted optimal trajectory will be achieved without exceeding the aircraft performance and control limits (i.e., ensure $\mathbf{U} \in \mathcal{U}$, $\mathbf{X} \in \mathcal{X}$), so $C^p = A_p e^{-\alpha_p d_p} / d_p$, where C^p is the aircraft performance constraint term added to the total cost function, A_p is the scaling factor, α_p is the decay rate and d_p is the performance margin given by:

$$d_p(\%) = 100 - 100 \left(\frac{\text{current state value}}{\text{state max} \setminus \text{min value}} \right) \quad (40)$$

To avoid a zero value of d_p , a minimum performance margin value d_{\min} must be defined so that:

$$\text{if } d_p \leq d_{\min} \text{ then } d_p = d_{\min}$$

The collision avoidance constraint, $\mathbf{Y} \in \mathcal{Y}$, is also represented by the Yukawa potential function and added to the cost function. As for the performance constraints, this punishes the cost function if the aircraft approaches an obstacle, so $C^{ob} = A_{ob}e^{-\alpha_{ob}d_{ob}}/d_{ob}$, where C^{ob} is a penalty term that represents the obstacle constraints, A_{ob} is a scaling factor, α_{ob} is the decay rate and d_{ob} is the distance between the nearest point on the obstacle and the point of interest.

Using potential functions to describe the obstacle constraints simplifies the search algorithm in the optimization process and handles the collision event in a manner which is closer to human behaviour. For example, avoidance manoeuvres can vary according to many factors such as aircraft speed, obstacle speed, aircraft manoeuvrability, and obstacle manoeuvrability. Additionally, due to the difficulty in generating a full 3D illustration for the obstacles that are detected by the on-board sensor unit the potential function approach does not need a 3D description of an obstacle, it just needs the distance between the aircraft and the nearest point in the obstacle [8].

7.3. Total Cost Function

The following cost function is used for the optimization process:

$$J = \sum_{i=1}^n \left[\lambda_p J_i^p + \lambda_s J_i^s + \lambda_{pc} J_i^{pc} + \lambda_{ob} J_i^{ob} \right] + \lambda_t J^t \quad (41)$$

where $J_i^p = (x_i^d - x_i^a)^2 + (y_i^d - y_i^a)^2 + (z_i^d - z_i^a)^2$ is the position cost function, $J_i^s = (u_i^d - u_i^a)^2 + (v_i^d - v_i^a)^2 + (w_i^d - w_i^a)^2$ is the speed cost function, $J_i^{pc} = \sum_{j=1}^q A_p e^{-\alpha_p d_p} / d_p$ is the vehicle performance constraints penalty function, $J_i^{ob} = \sum_{j=1}^m A_{ob} e^{-\alpha_{ob} d_{ob}} / d_{ob}$ is the obstacle constraints penalty function, and $J^t = \lambda_h (\psi_n^d - \psi_n^a)^2 + \lambda_f (\gamma_n^d - \gamma_n^a)^2$ is the final target cost function and where λ are scaling factors, n is the number of points that will be evaluated across the design horizon, q is the number of performance constraints, m is the number of detected obstacles, ψ is the heading angle and γ is the flight path angle. The superscript a means the actual value, while the superscript d means the demanded value.

The scaling factors λ can be tuned to control the balance between the different terms of the total cost function; trajectory tracking terms (J^p , J^s , J^t), and the constraints terms (the obstacle avoidance term J^{ob} and performance constraint term J^{pc}). The scaling factors can be constants or they may vary according to the situation. In other words, the priority of the cost function terms can be varied in order to allow the aircraft to fly safely in different flight scenarios.

The optimal control problem is solved using a gradient-based method, which suffers from the local minimum problem. The performance constraints tend to act as an enclosing boundary around the entire search space, hence are less likely to result in local minimum. However, when obstacles are detected this can have the impact of dividing the feasible design space into unconnected regions, therefore reducing the effectiveness of the solver of the optimal problem. Thus, the obstacle constraints are the primary source of the local minima. This paper proposes a method that reduces the possibility of getting trapped in local minimum by providing a mechanism for the search to jump to the different regions of the design space. A set of candidate trajectories are generated by applying maximum/minimum inputs to the vehicle model with the current vehicle states as initial states to ensure that the maximum performance manoeuvres in each axis are always available if required. Then the one that gives the minimum cost is used to initiate the optimal problem solver. In this case the input commands are:

$$\phi = [\phi_{\min} \quad \phi_c \quad \phi_{\max}], \quad T = [T_{\min} \quad T_c \quad T_{\max}], \quad n = [n_{\min} \quad n_c \quad n_{\max}] \quad (42)$$

where ϕ_c , T_c , and n_c are the current values of the inputs, and $\phi_{\min/\max}$, $T_{\min/\max}$, and $n_{\min/\max}$ are the minimum and maximum values of the inputs which can be calculated from the vehicle specifications (the Aerosonde UAV [63] model and specifications are used here). This combination will produce $3^3 = 27$ candidate trajectories.

8. Local Trajectory Planning Algorithm Simulation Results

The effectiveness of the proposed local trajectory planner is tested by simulating a set of collision scenarios. Figure 26 shows the system block diagram that is used in MATLAB/Simulink to produce the simulation results.

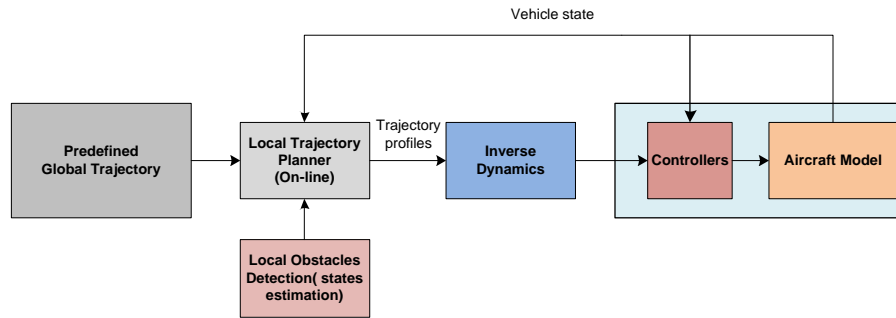


Figure 26. System block diagram.

The global trajectory is level flight with constant speed $v = 30$ m/s, altitude 1000 m and heading $\psi = 0$. The receding horizon time is $t_h = 20$ s and sampling time $t_s = 0.2$ s. The optimization process is updated every 2 s. The system is built in MATLAB/Simulink and the `fminunc` function is used as a solver for the unconstrained optimal control problem. The scaling factor values are: $\lambda_p = 100$, $\lambda_s = 500$, $\lambda_{pc} = 1$, $\lambda_{ob} = 1$, $\lambda_t = 1$, $\lambda_h = 10$, $\lambda_f = 1$. The obstacle is represented as a sphere, and a 4D model of the moving obstacle is generated using a straight projection method [64], which assumes that the obstacle does not manoeuvre during the receding horizon time.

8.1. Trajectory Tracking and Pop-up Obstacle Avoidance

The UAV initial speed and heading is matched to the global trajectory, but its initial position is vertically shifted from the global trajectory. A static pop-up obstacle must be avoided by the UAV. The simulation result presented in Figure 27 shows that the UAV is converging to the global trajectory before successfully avoiding the static obstacle. After passing the static obstacle, the UAV again converges to the global trajectory.

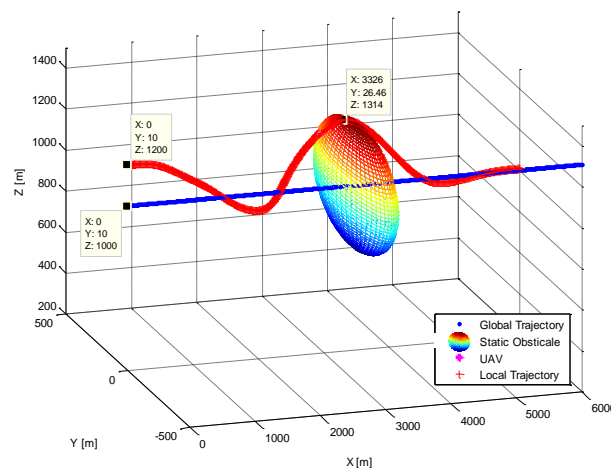


Figure 27. Converging to the global trajectory and avoiding a pop-up obstacle.

The UAV position, speed, heading angle, ψ , and flight path angle, γ , simulation results of this scenario are shown in Figure 28.

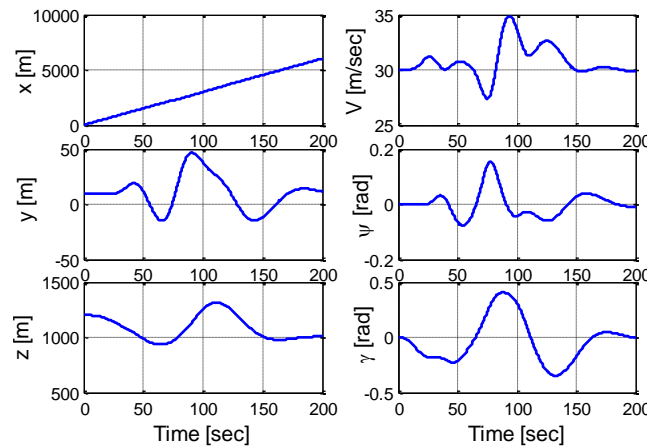


Figure 28. Position, speed, heading, and flight path angle during the manoeuvre.

8.2. Global Trajectory Tracking with Two Moving Intruders

This multiple conflict scenario simulates a situation where the UAV encounters two potential collisions, head-on and overtaking. The UAV is in level flight at the initial position (0,10,1000) m, initial heading $\psi = 0$, and initial constant speed $v = 30$ m/s. The first intruder (Intruder1) is in level flight at initial position (2000,10,1000) m, initial heading $\psi = \pi$ rad, and constant speed $v = 18$ m/s. The second intruder (Intruder2) is in level flight at initial position (2100,10,1000) m, heading $\psi = 0$ rad, constant speed $v = 15$ m/s. The UAV is in head-on collision risk with Intruder1 so it will then overtake Intruder2. The protection zone around each intruder is assumed to be 200 m. The UAV and the intruders paths during these scenarios are shown in Figure 29. Both collision scenarios have been avoided by the UAV successfully and then it returns to track the global trajectory after completing the overtaking manoeuvre around Intruder2. To clarify the performed manoeuvres, the projection of the UAV position on the horizontal and the vertical planes are included in Figure 29.

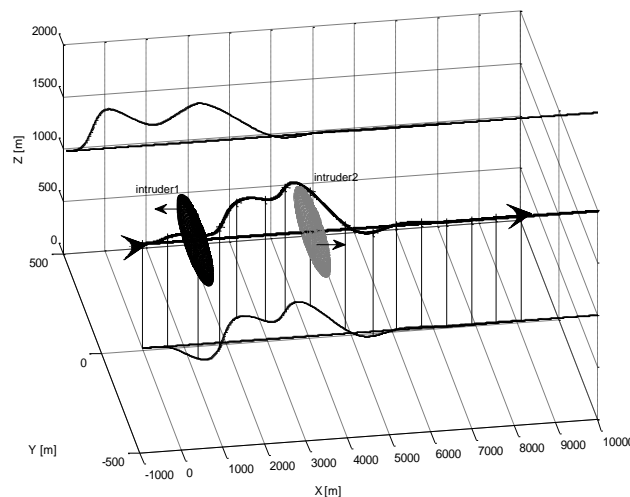


Figure 29. Collision avoidance scenarios, head-on (intruder1), overtaking (intruder2).

The intruders, with their protection zones shown as spheres, are presented in Figure 29 at the moment when they and the UAV have the same position on the x -axis. The time histories of UAV position, speed, heading angle ψ , and flight path angle γ during these scenarios are shown in Figure 30. The UAV and intruders' positions on the x -axis are shown in the top-left subplot in Figure 30 (UAV

(solid line), Intruder1 (dashed line), and Intruder2 (dotted line)), while the other two subplots in the left column show the y and z distance time histories. It can be noted that when the UAV and one of the intruders have the same x distance, y and z will be at their maximum values, so the UAV is avoiding a conflict with the intruders.

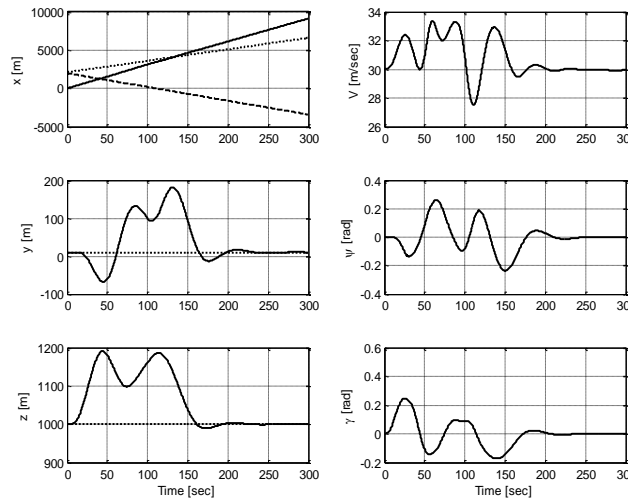


Figure 30. Position, speed, heading, and flight path during the manoeuvre.

9. Conclusions

A multi-layer DMS is developed for a sense and avoid system based on the rules of the air in VFR conditions. The proposed DMS architecture is engineered to be implemented for different functionalities for manned aircraft, and at different level of autonomy of UAS. The avoidance manoeuvres generating process for different conflict scenarios, in which the UAV should change direction (right/left turn) in horizontal plane, have been discussed. Two conflict scenarios are discussed, namely head-on, and right approaching. The rules of the air give general instructions for avoiding different conflict scenarios, but there are no specific procedures for performing the avoidance manoeuvres. Hence, the proposed avoidance manoeuvres that are presented in this paper are selected based on a pilot's suggestions. A geometric approach is used to parameterize the proposed avoidance manoeuvres. The avoidance manoeuvre that generated by the DMS can be used as a global trajectory for the developed optimal local trajectory planning LTP algorithm.

An optimal local trajectory generation that uses B-splines is proposed for a real-time collision avoidance algorithm. Online avoidance manoeuvre generation, optimization, and global trajectory tracking for different conflict scenarios are tested successfully in a simulation environment. The predicted trajectory is generated by using MPC techniques. Essentially, a finite-horizon optimal control problem is periodically solved in real-time hence updating the aircraft trajectory to avoid obstacles and track a predefined global trajectory. The aircraft and obstacle constraints are augmented in the cost function using a penalty function method. The computational time for the real-time collision avoidance algorithm is reduced significantly by using the output space to formulate the optimal control problem, and augmenting the vehicle/obstacle constraints in the cost function. A coarse grid approach is proposed to help the optimal control problem solver to escape the local minima and ensure sufficient coverage of the overall design space. Differential flatness of the system for a fixed wing aircraft is used to produce an inverse dynamic model for the UAV. Hence, the generated local trajectory profiles passed to the inverse dynamic model to generate the corresponding control signals. The simulation results show that the proposed approach allows the UAV to track a predefined global trajectory, as well as avoiding collisions with different types of conflict scenarios in real-time.

Author Contributions: Conceptualization, H.A. and J.W.; methodology, H.A.; software, H.A.; validation, H.A. and J.W.; formal analysis, H.A. and J.W.; investigation, H.A. and J.W.; resources, J.W.; data curation, H.A. and J.W.;

writing—original draft preparation, H.A. and J.W.; writing—review and editing, H.A. and J.W.; visualization, H.A. and J.W.; supervision, J.W.; project administration, H.A. and J.W.; funding acquisition, H.A. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. What Is the NURBS Curve

A NURBS curve is a vector-valued piecewise rational polynomial function. The p th degree NURBS curve is given by:

$$P(\tau) = \sum_{i=0}^n R_{i,p}(\tau) C_i \quad (A1)$$

$$R_{i,p}(\tau) = \frac{w_i N_{i,p}(\tau) C_i}{\sum_{i=0}^n w_i N_{i,p}(\tau)}; \quad a \leq \tau \leq b \quad (A2)$$

where $R_{i,p}(\tau)$ are rational basis functions. The analytical properties of $R_{i,p}(\tau)$ determine the geometric behavior of curves [65], w_i are the weights, C_i are the control points, and $N_{i,p}(\tau)$ are the p th degree B-spline basis functions. There are many ways to represent B-spline basis functions, for computer implementation the recursive representation of B-spline basis functions is the most useful form [65]. Let $U = [u_0, u_1, \dots, u_{m-1}, u_m]$ be a non-decreasing sequence of real numbers i.e., $u_i \leq u_{i+1}$; $i = 0, 1, \dots, m-1$, u_i called knots or breakpoints, and U is the knot vector that contain $m+1$ knots. So the i th B-spline basis function of p -degree (order $p+1$), denoted by $N_{p,i}(\tau)$ is defined as:

$$N_{i,0}(\tau) = \begin{cases} 1 & \text{if } u_i \leq \tau < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (A3)$$

$$N_{i,p}(\tau) = \frac{\tau - u_i}{u_{i+p} - u_i} N_{i,p-1}(\tau) + \frac{u_{i+p+1} - \tau}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(\tau) \quad (A4)$$

and $N_{i,p}(\tau) = 0$ if τ is outside $[u_i, u_{i+p+1}]$. The degree of the basis function p , number of control point $(n+1)$, and number of the knots $(m+1)$ are related by $m = n + p + 1$.

The knot vector can be realized in different forms, but it must be a nondecreasing sequence of real numbers. There are two types of knot vector, periodic and open, in two flavours, uniform and nonuniform [58]. In a uniform knot vector, individual knot values are evenly spaced. In practice, uniform knot vectors generally begin at zero and are incremented by 1 to some maximum value, or it can be normalized in a range between 0 and 1. A periodic uniform knot vector will give periodic uniform basis functions for which:

$$N_{i,p}(\tau) = N_{i-1,p}(\tau - 1) = N_{i+1,p}(\tau + 1) \quad (A5)$$

Thus, each basis function is a translation of the other. In an *open uniform* knot vector, the end knot values have multiplicity equal to the order of the B-spline basis functions $p+1$. NURBS basis functions have many useful properties [58]. For example, they are non-negative, satisfy the portion of unity property, have a local support, remain in the convex hull of the control points, and all their derivatives exist in the interior of the knot span $[u_i, u_{i+p+1}]$, where they are rational functions with nonzero denominators. The recursive calculation of the NURBS basis functions makes them easily, efficiently, and accurately processed in a computer.

The derivatives of B-spline curves can be calculated simply by computing the derivatives of their B-spline basis functions. The k th derivative of $P(\tau)$, $P^{(k)}(\tau)$, is given by:

$$P^{(k)}(\tau) = \sum_{i=0}^n N_{i,p}^{(k)}(\tau) C_i \quad (\text{A6})$$

where $N_{i,p}^{(k)}(\tau)$ is the k th derivative of B-spline basis functions which can be calculated recursively:

$$N_{i,p}^{(k)}(\tau) = p \left(\frac{N_{i,p-1}^{(k-1)}(\tau)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}(\tau)}{u_{i+p+1} - u_{i+1}} \right) \quad (\text{A7})$$

Bezier curves represent a special case of NURBS where all the weights are equal to unity, i.e., $w_i = 1$, and the knot vector is $U = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]$ (for $p = 6$). In this case the basis functions are called Bernstein basis functions.

References

1. Lai, C.K.; Lone, M.; Thomas, P.; Whidborne, J.F.; Cooke, A.K. On-board trajectory generation for collision avoidance in unmanned aerial vehicles. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2011; pp. 1–14. [10.1109/AERO.2011.5747526](#). [[CrossRef](#)]
2. U.S. Department of Transportation. *Integration of Civil Unmanned Aircraft Systems (UAS) in The National Airspace System (NAS) Roadmap*; Technical Report; U.S. Department of Transportation, FAA: Washington, DC, USA, 2013.
3. Hutchings, T.; Jeffries, S.; Farmer, S. Architecting UAV sense and avoid systems. In Proceedings of the IET Conference on Autonomous Systems, London, UK, 8 February 2007; pp. 1–8.
4. Pellebergs, J. The MIDCAS Project. In Proceedings of the 27th International Congress of the Aeronautical Sciences, Nice, France, 19–24 September 2010.
5. Osinga, F.P. *Science, Strategy and War: The Strategic Theory of John Boyd*; Routledge: London, UK, 2006.
6. U.S. Department of Transportation. *Advisory Circular: Pilots' Role in Collision Avoidance*; Technical Report 90-48C; Federal Aviation Administration: Washington, DC, USA, 1983.
7. Australian Transport Safety Bureau. *Limitations of the See-and-Avoid Principle*; Australian Transport Safety Bureau (ATSB): Adelaide, Australia, 2004.
8. Berry, A.; Howitt, J.; Gu, D.; Postlethwaite, I. Continuous Local Motion Planning & Control for Micro-Air-Vehicles in Complex Environments. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Toronto, ON, Canada, 2–5 August 2010; Number AIAA 2010-7874. [10.2514/6.2010-7874](#). [[CrossRef](#)]
9. Alturbeh, H.; Whidborne, J.F. Real-time Obstacle Collision Avoidance for Fixed Wing Aircraft Using B-splines. In Proceedings of the UKACC International Conference in Control (Control2014), Loughborough, UK, 9–11 July 2014; doi:[10.1109/CONTROL.2014.6915125](#). [[CrossRef](#)]
10. Zeitlin, A. Sense and Avoid capability development challenges. *IEEE Aerosp. Electron. Syst. Mag.* **2010**, *25*, 27–32. [10.1109/MAES.2010.5631723](#). [[CrossRef](#)]
11. Kuchar, J.; Yang, L. A review of conflict detection and resolution modeling methods. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 179–189. [10.1109/6979.898217](#). [[CrossRef](#)]
12. Albaker, B.; Rahim, N. A survey of collision avoidance approaches for unmanned aerial vehicles. In Proceedings of the International Conference for Technical Postgraduates (TECHPOS), Kuala Lumpur, Malaysia, 14–15 December 2009; pp. 1–7. [10.1109/TECHPOS.2009.5412074](#). [[CrossRef](#)]
13. Breen, B. Controlled Flight Into Terrain and the enhanced Ground Proximity Warning system. *IEEE Aerosp. Electron. Syst. Mag.* **1999**, *14*, 19–24. [10.1109/62.738350](#). [[CrossRef](#)]
14. Tomlin, C.; Pappas, G.; Sastry, S. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Trans. Autom. Control* **1998**, *43*, 509–521. [10.1109/9.664154](#). [[CrossRef](#)]

15. Wollkind, S.; Valasek, J.; Ioerger, T.R. Automated conflict resolution for air traffic management using cooperative multiagent negotiation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; Volume 2, pp. 1078–1088.
16. Sislak, D.; Rehak, M.; Pechoucek, M.; Pavlicek, D.; Uller, M. Negotiation-Based Approach to Unmanned Aerial Vehicles. In Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006), Prague, Czech Republic, 26 June 2006; pp. 279–284.10.1109/DIS.2006.55. [[CrossRef](#)]
17. Hill, J.C.; Johnson, F.R.; Archibald, J.K.; Frost, R.L.; Stirling, W.C. A cooperative multi-agent approach to free flight. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, 25–29 July 2005; pp. 1083–1090.
18. U.S. Department of Transportation. *Introduction to TCAS II Version 7.1*; FAA: Washington, DC, USA, 2011.
19. Shim, D.; Sastry, S. A situation-aware flight control system design using real-time model predictive control for unmanned autonomous helicopters. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; Volume 2, pp. 855–862.10.2514/6.2006-6101. [[CrossRef](#)]
20. Shim, D.; Chung, H.; Sastry, S. Conflict-free navigation in unknown urban environments. *Robot. Autom. Mag.* **2006**, *13*, 27–33.10.1109/MRA.2006.1678136. [[CrossRef](#)]
21. Shim, D.; Kim, H.; Sastry, S. Decentralized nonlinear model predictive control of multiple flying robots. In Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; Volume 4, pp. 3621–3626.10.1109/CDC.2003.1271710. [[CrossRef](#)]
22. Dong, T.; Liao, X.H.; Zhang, R.; Sun, Z.; Song, Y. Path Tracking and Obstacles Avoidance of UAVs-Fuzzy Logic Approach. In Proceedings of the 14th IEEE International Conference on Fuzzy Systems, Reno, NV, USA, 25 May 2005; pp. 43–48.
23. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; Volume 2, pp. 500–505.10.1109/ROBOT.1985.1087247. [[CrossRef](#)]
24. Goss, J.; Rajvanshi, R.; Subbarao, K. Aircraft conflict detection and resolution using mixed geometric and collision cone approaches. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; Volume 1, pp. 670–689.
25. Chakravarthy, A.; Ghose, D. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Trans. Syst. Man and Cybern. Part A Syst. Hum.* **1998**, *28*, 562–574.10.1109/3468.709600. [[CrossRef](#)]
26. Alsaab, A.; Bicker, R. Behavioral Strategy for Indoor Mobile Robot Navigation in Dynamic Environments Navigation in Dynamic Environments. *Int. J. Eng. Sci. Innov. Technol. (IJESIT)* **2014**, *3*, 533–542.
27. Dobrokhodov, V.; Kaminer, I.; Jones, K.; Ghabcheloo, R. Vision-based tracking and motion estimation for moving targets using small UAVs. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.10.1109/ACC.2006.1656418. [[CrossRef](#)]
28. Tomlin, C.; Lygeros, J.; Sastry, S. A game theoretic approach to controller design for hybrid systems. *Proc. IEEE* **2000**, *88*, 949–970.10.1109/5.871303. [[CrossRef](#)]
29. Cowling, I. Towards Autonomy of a Quadrotor UAV. Ph.D. Thesis, Cranfield University, School of Engineering, Cranfield, UK, 2008.
30. Latombe, J.C. *Robot Motion Planning*; Springer: New York, NY, USA, 1991; doi:10.1007/978-1-4615-4022-9. [[CrossRef](#)]
31. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.10.1017/CBO9780511546877. [[CrossRef](#)]
32. Sigurd, K.; How, J. UAV trajectory design using total field collision avoidance. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003.10.2514/6.2003-5728. [[CrossRef](#)]
33. Smith, A.L.; Harmon, F.G. UAS collision avoidance algorithm minimizing impact on route surveillance. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Chicago, IL, USA 10–13 August 2009.
34. Barraquand, J.; Langlois, B.; Latombe, J.C. Numerical potential field techniques for robot path planning. In Proceedings of the Fifth International Conference on Advanced Robotics: ‘Robots in Unstructured Environments, Pisa, Italy, 19–22 June 1991; Volume 2, pp. 1012–1017.10.1109/ICAR.1991.240539. [[CrossRef](#)]

35. Barraquand, J.; Latombe, J.C. A Monte-Carlo algorithm for path planning with many degrees of freedom. In Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; Volume 3, pp. 1712–1717.10.1109/ROBOT.1990.126256. [\[CrossRef\]](#)
36. Barraquand, J.; Latombe, J.C. Robot Motion Planning: A Distributed Representation Approach. *Int. J. Robot. Res.* **1991**, *10*, 628–649.10.1177/027836499101000604. [\[CrossRef\]](#)
37. Barraquand, J.; Kavraki, L.; Latombe, J.C.; Motwani, R.; Li, T.Y.; Raghavan, P. A Random Sampling Scheme for Path Planning. *The Int. J. Robot. Res.* **1997**, *16*, 759–774.10.1177/027836499701600604. [\[CrossRef\]](#)
38. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580.10.1109/70.508439. [\[CrossRef\]](#)
39. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271.10.1007/BF01386390. [\[CrossRef\]](#)
40. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107.10.1109/TSSC.1968.300136. [\[CrossRef\]](#)
41. Lavalley, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report; Iowa State University: Ames, IA, USA, 1998.
42. Lavalley, S.M.; Kuffner, J.J. Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics: New Directions*; CRC Press: Boca Raton, FL, USA, 2000; pp. 293–308.
43. LaValley, S.M.; Kuffner, J.J. Randomized Kinodynamic Planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400.10.1177/02783640122067453. [\[CrossRef\]](#)
44. Ebdon, D.; Regan, J. *White Paper: Sense-and-Avoid Requirement for Remotely Operated Aircraft (ROA)*; HQ ACC/DR-UAV SOM: 2005. Available online: <https://www.yumpu.com/en/document/view/38520835/white-paper-amtech-usaorg> (accessed on 24 February 2020).
45. Euteneuer, E.A.; Papageorgiou, G. UAS insertion into commercial airspace: Europe and US standards perspective. In Proceedings of the IEEE/AIAA 30th Digital Avionics Systems Conference (DASC), Seattle, WA, USA, 16–20 October 2011; pp. 5C5:1–5C5:12.10.1109/DASC.2011.6096084. [\[CrossRef\]](#)
46. Safety Regulation Group. *Unmanned Aircraft System Operations in UK Airspace-Guidance*; Technical Report CAP-722; Civil Aviation Authority: West Sussex, UK, 2012.
47. Gimenes, R.A.V.; Correa, M.A.; Camargo, J.B.; Avelino, V.F.; Vismari, L.F.; Cugnasca, P.S.; Rossi, M.A.; Almeida, J.R. Guidelines for integration of autonomous UAS in Global ATM. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 994–1003.10.1109/ICUAS.2013.6564787. [\[CrossRef\]](#)
48. Valavanis, K.P.; Vachtsevanos, G.J., UAV Integration into the National Airspace: Introduction. In *Handbook of Unmanned Aerial Vehicles*; Springer: Dordrecht, The Netherlands, 2015; pp. 2113–2116.10.1007/978-90-481-9707-1_147. [\[CrossRef\]](#)
49. Shakernia, O.; Chen, W.Z.; Graham, S.; Zvanya, J.; White, A.; Weingarten, N.; Raska, V.M. Sense and Avoid (SAA) Flight Test and Lessons Learned. In Proceedings of the AIAA Infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, CA, USA, 7–10 May 2007; pp. 1–15.10.2514/6.2007-3003. [\[CrossRef\]](#)
50. Dubot, T. Integrating civil unmanned aircraft operating autonomously in non-segregated airspace: Towards a dronoethics? In Proceedings of the ECAI Workshop on Rights and Duties of Autonomous Agents, Montpellier, France, 28 August 2012; Volume 885.
51. Clemens, J. *Portable Collision Avoidance System (PCAS) Model XRX Owners Manual*; Zaon Flight Systems, Inc.: Frisco, TX, USA, 2005.
52. Thom, T. *Air Pilot's Manual-Aviation Law & Meteorology*; Airline Publishing: Shrewsbury, UK, 2004; Volume 2.
53. Spence, C. *AIM/FAR 2008: Aeronautical Information Manual/Federal Aviation Regulations*; McGraw-Hill Professional: New York, NY, USA, 2007.
54. ICAO. Annex 2 to the convention on international civil aviation: Rules of the Air. In *International Standards*; ICAO: Montreal, QC, Canada, 2005.
55. Stevens, B.L.; Lewis, F.L. *Aircraft Control and Simulation*, 2nd ed.; John Wiley: Hoboken, NJ, USA, 2003.
56. Cook, M.V. *Flight Dynamics Principles*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2007.
57. Yang, K.; Kang, Y.; Sukkarieh, S. Adaptive nonlinear model predictive path-following control for a fixed-wing unmanned aerial vehicle. *Int. J. Control Autom. Syst.* **2013**, *11*, 65–74.10.1007/s12555-012-0028-y. [\[CrossRef\]](#)
58. Piegl, L.; Tiller, W. *The NURBS Book*; Springer: New York, NY, USA, 1997.

59. Cowling, I.D.; Yakimenko, O.A.; Whidborne, J.F.; Cooke, A.K. Direct Method Based Control System for an Autonomous Quadrotor. *J. Intell. Robot. Syst.* **2010**, *60*, 285–316. [[CrossRef](#)]
60. Sira-Ramírez, H.; Agrawal, S.K. *Differentially Flat Systems*; Marcel Dekker: New York, NY, USA, 2004.
61. Berry, A. Continuous Local Motion Planning and Control for Unmanned Vehicle Operation within Complex Obstacle-Rich Environments. Ph.D. Thesis, University of Leicester, Leicester, UK, 2010.
62. Cohen-Tannoudji, C.; Diu, B.; Laloe, F. *Quantum Mechanics*; Wiley: Hoboken, NJ, USA, 1977.
63. Holland, G.J.; McGeer, T.; Youngren, H. Autonomous Aerosondes for Economical Atmospheric Soundings Anywhere on the Globe. *Bull. Amer. Meteor. Soc.* **1992**, *73*, 1987–1998.<1987:AAFEAS>2.0.CO;2. [[CrossRef](#)]
64. Albaker, B.; Rahim, N. Straight projection conflict detection and cooperative avoidance for autonomous unmanned aircraft systems. In Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009), Xi'an, China, 25–27 May 2009; pp. 1965–1969.10.1109/ICIEA.2009.5138546. [[CrossRef](#)]
65. Piegl, L. On NURBS: A survey. *Comput. Gr. Appl.* **1991**, *11*, 55–71. doi:10.1109/38.67702. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

2020-02-25

Visual flight rules-based collision avoidance systems for UAV flying in civil aerospace

Alturbeh, Hamid

MDPI

Alturbeh H, Whidborne JF. (2020) Visual flight rules-based collision avoidance systems for UAV flying in civil aerospace. *Robotics*, Volume 9, February 2020, Article number 9

<https://doi.org/10.3390/robotics9010009>

Downloaded from Cranfield Library Services E-Repository